

Securing and Auditing Data on DB2 for z/OS

Prepare for the threat from within and
without

Comply with IBM Data Server
Security Blueprint

Extend the skills of data
professionals



Paolo Bruni
Felipe Bortoletto
Thomas Hubbard
Ernest Mancill
Hennie Mynhardt
Shuang Yu

Redbooks



International Technical Support Organization

Securing and Auditing Data on DB2 for z/OS

June 2009

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

First Edition (June 2009)

This edition applies to Version 9.1 of IBM DB2 for z/OS (program number 5635-DB2), Version 2.1 of IBM DB2 Audit Management Expert for z/OS (program number 5655-I16), and Version 1.1 of IBM Data Encryption for IMS and DB2 Databases (program number 5655-P03).

© Copyright International Business Machines Corporation 2009. 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
Examples	xiii
Tables	xv
Notices	xvii
Trademarks	xviii
Preface	xix
The team that wrote this book	xix
Become a published author	xxi
Comments welcome	xxi
Part 1. Data governance	1
Chapter 1. Regulatory compliance	3
1.1 Recent events	4
1.2 IBM data governance roadmap	5
1.2.1 Data Governance Council	6
1.3 Regulations	7
1.3.1 Payment Card Industry Data Security Standard (PCI DSS)	7
1.3.2 Basel II	13
1.3.3 Gramm-Leach-Bliley Act	14
1.3.4 Health Insurance Portability and Accountability Act	14
1.3.5 California Security Breach Information Act	15
1.3.6 Sarbanes-Oxley Act	15
Chapter 2. The IBM Data Server security roadmap and some common DB2 for z/OS security themes	19
2.1 The IBM Data Server Security Blueprint	20
2.1.1 Introduction and overview	20
2.1.2 Why a Data Server Security Blueprint?	21
2.1.3 Invest in the future	23
2.2 Threat elements of the IBM Data Server Security Blueprint	24
2.2.1 Data security layers	24
2.2.2 Data threats	24
2.2.3 Configuration threats	26
2.2.4 Audit threats	26
2.2.5 Executable threats	26
2.3 Threat countermeasures	27
2.3.1 Data threats	29
2.3.2 Configuration treats	32
2.3.3 Audit threats	33
2.3.4 Executable threats	33
2.4 Interpretation of some DB2 for z/OS common security themes	34
2.4.1 Separation of roles	34
2.4.2 Audit versus external security	36
2.4.3 Personally identifying information and index encryption	38
2.4.4 Encryption standards	41

2.4.5	Cost of security versus SLA	42
2.4.6	The cost of a data breach	43
2.4.7	ROI calculation	45
Part 2.	IBM data governance portfolio	47
Chapter 3.	IBM data servers on z/OS	49
3.1	Security categorization	50
3.1.1	Data servers security areas	50
3.2	DB2	51
3.2.1	Authentication	51
3.2.2	Authorization	57
3.2.3	SQL	67
3.2.4	Application security	68
3.2.5	Encryption	68
3.2.6	Network security	73
3.2.7	Auditing	74
3.3	IMS	78
3.3.1	Authorization	78
3.3.2	Encryption	80
3.3.3	Auditing	82
3.4	VSAM	85
3.4.1	Authorization	86
3.4.2	Encryption	86
3.4.3	VSAM auditing	89
Chapter 4.	IBM information management tools	91
4.1	DB2 Audit Management Expert for z/OS	92
4.2	Data Encryption for IMS and DB2 Databases Tool	93
4.2.1	DB2 encryption	93
4.2.2	IMS encryption	94
4.2.3	Data Encryption for IMS and DB2 Databases Tool summary	94
4.3	Log Analysis Tool	95
4.4	Performance tools	95
4.4.1	DB2 Query Monitor	96
4.4.2	Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS	96
Chapter 5.	Tivoli products	99
5.1	Tivoli zSecure suite	100
5.1.1	zSecure Administration products	100
5.1.2	zSecure Audit Products	103
5.2	Tivoli Security Information and Event Manager	105
5.2.1	Tivoli Compliance Insight Manager	105
5.2.2	Tivoli Security Operations Manager	106
5.2.3	The combined value	106
Chapter 6.	Optim solutions	109
6.1	Introduction	110
6.2	IBM Optim Data Growth Solution for z/OS	111
6.3	IBM Optim Data Privacy Solution	113
6.4	IBM Optim Test Data Management Solution	116
6.5	IBM Optim Database Relationship Analyzer	120
Part 3.	System z synergy	125

Chapter 7. System z security features	127
7.1 System z integrated cryptography	128
7.1.1 Cryptographic hardware	128
7.1.2 IBM Common Cryptographic Architecture	132
7.1.3 Logical partitioning and System z hardware cryptography exploitation	138
7.1.4 Monitoring the cryptographic workload on z/OS	139
7.1.5 Sysplex and System z hardware cryptography	140
7.1.6 Software requirements	140
7.1.7 ICSF bibliography	141
7.2 DS8000—Encrypting disk storage	141
7.3 TS1120—Encrypting tape storage	143
7.4 zIIP	148
7.4.1 IPsec encryption and zIIP exploitation	148
7.4.2 zIIP and Encryption Tool for IMS and DB2 Databases	149
Chapter 8. z/OS security	151
8.1 Integrated Cryptographic Service Facility	152
8.1.1 Middleware ICSF exploitation	152
8.1.2 Resource Access Control Facility	154
8.2 Communication Server	157
8.3 z/OS Encryption Facility	159
Part 4. DB2 Audit Management Expert	161
Chapter 9. DB2 Audit Management Expert architecture and installation	163
9.1 Architectural overview	164
9.1.1 General functions	165
9.1.2 Components	166
9.2 Storage modes	168
9.2.1 Load repository mode	168
9.2.2 Generate off load data sets mode	169
9.2.3 Dual mode	171
9.3 Installation and configuration	172
9.3.1 Planning for the installation	172
9.4 Security	174
9.5 XML	175
9.6 Data sharing	175
9.7 Installing and configuring DB2 Audit Management Expert for z/OS	176
Chapter 10. Audit Management Expert scenarios	211
10.1 Defining audit responsibilities	212
10.2 Reporting User Interface	222
10.2.1 Introduction to Reporting User Interface	222
10.2.2 Auditing privileged users	228
10.2.3 Finding all authorization failures	242
10.2.4 Finding DDL activity	252
10.3 Log Analysis User Interface	257
10.3.1 Generating Log Analysis reports	257
10.3.2 Templates and jobs	267
Chapter 11. Audit Management Expert administration	271
11.1 Separation of roles	272
11.2 Control (DBA versus auditor)	272
11.3 Performance monitoring	272

11.3.1	How to collect audit data	272
11.3.2	Controlling data collection	273
11.4	Repository administration	274
Part 5.	Data Encryption for IMS and DB2 Databases Tool	275
Chapter 12.	Architecture and ICSF key management	277
12.1	Integrated Cryptographic Service Facility	278
12.2	CEX2C configuration (HMC)	281
12.3	DES master key generation	288
12.3.1	Loading cryptographic processors with DES master key	288
12.3.2	PPINIT and CKDS initialization	292
12.3.3	HCR7751 and CKDS operations without CEX2C	295
Chapter 13.	Data Encryption tool installation and customization	299
13.1	Generation of an encryption EDITPROC	300
13.1.1	Generate a Clear Key using ICSF	300
13.2	DB2 encryption implementation scenario for the DBA	311
13.2.1	Creating the DB2 user exit routine by using ISPF panels	311
13.2.2	Implementing DB2 encryption	314
13.2.3	Max record size	314
Chapter 14.	Data encryption scenarios	315
14.1	Master key protected CKDS	316
14.1.1	Clear key	316
14.1.2	Encryption from a data management perspective	320
14.1.3	Encryption confirmation techniques	328
14.1.4	Secure key	330
14.1.5	AES 128 clear key	332
14.2	Clear-key-only Cryptographic Key Data Set (HCR7751)	333
14.3	Compression and encryption	334
14.3.1	Compression support in Data Encryption for IMS and DB2 Databases Tool	335
14.3.2	Additional encryption considerations with compressed data	341
14.3.3	Compression scenario	342
Chapter 15.	Administration of encrypted objects	345
15.1	Backup and recovery (local site considerations)	346
15.2	Disaster recovery considerations	346
15.3	Key rotation	349
15.4	Alteration of encrypted table schema	350
15.5	Failure scenarios	355
15.5.1	Key label mismatch in EDITPROC	355
15.5.2	CKDS failure - Master key mismatch	357
15.5.3	Out-of-synch key labels	360
15.6	Performance measurements	360
15.6.1	Utilities	361
15.6.2	SQL	363
Part 6.	Appendixes	365
Appendix A.	System topology and workload	367
A.1	Hardware and software set up	368
A.2	DB2 workload	368
A.2.1	Getting started - Installation instructions	370

Appendix B. Sample configuration files for DB2 Audit Management Expert for z/OS . .	
383	
B.1 Server configuration file	384
B.2 Agent configuration file	389
B.3 Audit SQL collector configuration file	394
Related publications	397
IBM Redbooks	397
Other publications	398
Online resources	398
How to get Redbooks	399
Help from IBM	399
Abbreviations and acronyms	401
Index	403

Figures

2-1	The Data Server Security Blueprint	20
2-2	The IBM Data Server Security Blueprint	28
2-3	Exposed index representation	39
2-4	Indirect index access representation	40
2-5	ROI Calculator	46
3-1	Associating IDs to process	58
3-2	DB2 privileges	61
5-1	zSecure Admin main menu	101
5-2	zSecure Visual GUI interface	102
5-3	zSecure Alert data flow	104
5-4	zSecure Command Verifier: Output from non-compliant RACF commands	104
5-5	TSIEM Solution	107
6-1	The Optim solutions	110
6-2	Optim data growth archiving	112
6-3	Optim Data Privacy data masking techniques	116
6-4	IBM Optim Test Data Management in action	117
6-5	IBM Optim improves every stage of the application testing process	119
6-6	Need for a way to capture relationally consistent data	120
6-7	Dealing with relationships	121
7-1	A CPACF is associated with every CP	129
7-2	Layout of a CEX2 feature	130
7-3	Processing inside the CEX2C during an encryption request	134
7-4	Overview of how the hardware and software work together	135
7-5	DS8000 Encryption Depiction	142
7-6	TS1120 tape encryption process flow	145
7-7	LTO4 Tape Encryption process	146
8-1	IPSec zIIP processing	159
8-2	Encryption services and clients	160
9-1	Load repository mode architecture	169
9-2	Generate Off load data set mode architecture	170
9-3	Dual mode architecture	171
9-4	Audit Management Expert Administration: Welcome	202
9-5	Audit Management Expert Administration: Choose Directory	203
9-6	Audit Management Expert Administration: Shortcut	203
9-7	Audit Management Expert Administration: Installation Summary	204
9-8	Audit Management Expert Administration: Installing	204
9-9	Audit Management Expert Administration: Installation Finish	205
9-10	Audit Management Expert Administration: Defining Server	205
9-11	Audit Management Expert Administration: Defining Server	206
9-12	Audit Management Expert Administration: Logging in	206
9-13	Audit Management Expert Report: Welcome	207
9-14	Audit Management Expert Report: Choose Directory	207
9-15	Audit Management Expert Report: Create Shortcut	208
9-16	Audit Management Expert Report: Installation summary	208
9-17	Audit Management Expert Report: Installation finish	209
9-18	Audit Management Expert Report: Defining Server entry	209
9-19	Audit Management Expert Report: Defining Server value	210
9-20	Audit Management Expert Report: Log in	210

10-1	DB2 Audit Management Expert Login	212
10-2	User administration list	213
10-3	New user wizard	214
10-4	User permissions	215
10-5	User group assignments	216
10-6	User Summary	217
10-7	User list	218
10-8	Setup user authorizations	219
10-9	Authorization editor	220
10-10	Authorizations List	221
10-11	Level1 report: Subsystem overview	223
10-12	Level2 report: Summary report for subsystem	224
10-13	Level3 report: Subsystem detail	225
10-14	Level3 report: Summary of objects in subsystem	226
10-15	Select the date range to report	227
10-16	Report filter for the subsystem and Level3 report	228
10-17	Overview of the audited DB2 subsystem	230
10-18	Report filter for subsystem	231
10-19	Exclude plans in Report Filter window	232
10-20	Overview of the audited subsystem with filter applied	233
10-21	Summary report for the audited DB2 subsystem	234
10-22	Change of audited object for the DB2 subsystem	235
10-23	Level3 report for the successful change of audited object	235
10-24	Statement executed against the audited object	236
10-25	Save report	236
10-26	Save report for later retrieval	237
10-27	Overview of the audited DB2 subsystem	237
10-28	Exclude the production authorized plan in the filter panel	238
10-29	Overview of audited subsystem with filter applied	239
10-30	Summary report of the audited subsystem	240
10-31	Read of audited objects for the subsystem	241
10-32	Level3 report of audited object read	241
10-33	Overview of monitored DB2 subsystem (DB9A) on system SC63	243
10-34	Selecting failures on DB2 subsystem (DB9A).	244
10-35	Access attempts failure	245
10-36	Summary Report of failure activity	246
10-37	Access attempt	247
10-38	Access attempt - detail information	247
10-39	Summary Report of failure activity	248
10-40	Other authorization failure window	249
10-41	Command failure window	249
10-42	Summary Report of failure activity	250
10-43	Read of audited object window	251
10-44	SQL failure window	251
10-45	SQL statement	252
10-46	DB2 SYSTEM overview	253
10-47	Activity overview	254
10-48	CREATE, ALTER, and DROP activity	255
10-49	List of executed DDL	255
10-50	DROP detail	256
10-51	List of executed DDL	256
10-52	CREATE detail	256
10-53	Log Analysis advisor "Welcome" window	258

10-54	Select Subsystem for Log Analysis	259
10-55	Generate User List	260
10-56	Add table list	261
10-57	Add statement	262
10-58	Generate JCL	263
10-59	Summary report	264
10-60	Detail report	265
10-61	View log	266
10-62	Open Template	267
10-63	Save Template	268
10-64	Delete Template	268
12-1	Visual representation of secure key	279
12-2	Visual representation of clear key	280
12-3	LPAR and domain relationship	282
12-4	ICSF startup without CEX2C available	282
12-5	Coprocessor management panel with no assigned CEX2C	283
12-6	HMC Console workplace	284
12-7	CPC Images Workarea - LPAR Designation	285
12-8	Activation profiles list	285
12-9	Control Domain/Usage Domain Assignment	286
12-10	Control Domain Cryptographic Candidate designation	287
12-11	ICSF coprocessor hardware status	288
12-12	IDCAMS VSAM Define statements for CKDS	289
12-13	ICSF JCL procedure	290
12-14	ICSF z/OS Start Command	290
12-15	ICSF startup parameters	290
12-16	ICSF Coprocessor Hardware Status	292
12-17	ICSF Startup messages with CEX2C available but no keys loaded	293
12-18	ICSF Main ISPF Menu	293
12-19	PPINIT Master Key and CKDS Initialization	294
12-20	Coprocessor Hardware status - After PPINIT	295
12-21	Master Key Management menu	296
12-22	CKDS Initialization - No CEX2C	297
13-1	Option 8 - Key Generator Utility processes	301
13-2	Key Utility Generator creation panel	302
13-3	Specifying the KGUP data set name	302
13-4	Example of input fields to specify KGUP data set name and attributes	303
13-5	Updating the KGUP data set	304
13-6	Creating and storing the key control statements	305
13-7	Help for the KEY field - all the key options	306
13-8	Creating the KGUP Key statement	307
13-9	Successful Update of the key	308
13-10	Contents of the Control file - showing the key label	308
13-11	Contents of the updated KGUP Control file using IBM File Manager for z/OS	309
13-12	Creating other needed data sets for the KGUP	309
13-13	Specify the names of the data sets needed for KGUP processing	310
13-14	JCL generated by ICSF for KGUP	310
13-15	Main menu for Data Encryption for IMS and DB2 Databases Tool	311
13-16	Data Encryption for IMS and DB2 Databases Tool Main Menu	312
13-17	Generated JCL which builds the encryption exit	313
14-1	KGUP clear key specification	316
14-2	Modified DECDB2CK from SDECSAMP	317
14-3	Insert processing with encrypting EDITPROC	320

14-4	Table definition used in the log scenario	322
14-5	Display of DB9A Log RBA status	322
14-6	DSN1LOGP Summary report JCL and Control Statements	323
14-7	Output from DSN1LOGP summary report	323
14-8	DSN1LOGP with LUWID for detail reporting.	324
14-9	DSN1LOGP UNDO/REDO of an encrypted table.	324
14-10	DSN1LOGP UNDO/REDO of a cleartext table	324
14-11	DDL for encrypted table - original	326
14-12	DDL for encrypted table - clone	326
14-13	SQL to determine object identifiers.	327
14-14	DSN1COPY JCL example	327
14-15	DSNTEP2 output from SELECT	328
14-16	DSN1PRNT JCL to print data pages of VSAM LDS	328
14-17	DSN1PRNT unencrypted text example	329
14-18	DSN1PRNT encrypted data page example	329
14-19	DSN1PRNT using image copy input.	330
14-20	DSN1PRINT encrypted image copy page	330
14-21	Secure key KGUP example	331
14-22	Secure key JCL sample	332
14-23	KGUP CLRAES key specification	333
14-24	DSN1COMP output example	337
14-25	Uncompressed sample tablespace - DB2 Administration Tool.	342
14-26	DSN1COMP Compression Dictionary built output	343
14-27	Editproc compressed table - DB2 Administration Tool	344
14-28	DB2 COMPRESS YES table - DB2 Administration Tool.	344
15-1	ICSF Coprocessor Hardware Status.	348
15-2	Administration Tool Alter Table.	351
15-3	DB2 Administration Tool - Add Column dialog	352
15-4	SQL Code -668 raised by ALTER TABLE	353
15-5	Valid keylabel ZAP output.	355
15-6	Updated key ZAP output.	356
15-7	S0C7 abend using incorrect clear key	357
15-8	ICSF startup with CKDS master key mismatch.	358
15-9	CKDS header record.	359
15-10	ICSF Coprocessor Hardware Status.	359
15-11	SQLCODE with CKDS mismatch	360
15-12	Traces active on DB9A	361
A-1	The GLW database	369

Examples

3-1	At the DB2 server	56
3-2	Defining a role	63
3-3	Giving USRT060 SELECT access on table EMP	66
3-4	Using Data Definition Language	67
3-5	Using a VIEW basic to standard	67
3-6	DB2 data encryption	70
3-7	Applying the DECRYPT_CHAR function	71
3-8	EMPNO decrypted	71
3-9	Encrypting data at the cell level	71
3-10	GETHINT information returned	71
3-11	Start trace with including or excluding roles	75
3-12	Alter table for auditing	75
3-13	Using OMEGAMON PE for audit trace	76
3-14	Audit trace sample output	76
3-15	Encipher using a Clear Key in REPRO command	87
3-16	Report from encipher JCL	87
3-17	Decipher using a Clear Key in REPRO command	87
3-18	Encipher using an encrypted key	88
3-19	Decipher using an encrypted key	88
3-20	Output of decipher using an encrypted key	88
7-1	Sample ICSF Coprocessor Management panel	131
7-2	Linking an ICSF callable service into an application program	138
9-1	Server configuration file	173
9-2	Agent configuration file	174
9-3	ADHEMAC1 macro customized	180
9-4	ADHSJ000: Create Audit Management Expert control file	181
9-5	ADHSJ001: Configure Audit Management Expert control file	182
9-6	ADHDDLA: Create Audit Management Expert audit repository	184
9-7	ADHDDL: Create Alias	185
9-8	ADHBND90: Bind Audit Management Expert packages for the repository	188
9-9	ADHBND91: Bind Audit Management Expert packages for Log Analysis and data collection	191
9-10	ADHGRT9B: Grant Audit Management Expert packages	194
9-11	ADHBND92: Bind Audit Management Expert plans	195
9-12	ADHCFG: Server configuration file	196
9-13	ADHSJSRV—Audit Management Expert server JCL	197
9-14	ADHINTER: Creating ADHINTER dataset	197
9-15	ADHCFG: ADH#MAIN program parameters	198
9-16	ADHCDB9A: ASC started task	198
9-17	DB9ACFGA	199
9-18	ADHDB9AA: Audit Management Expert agent JCL	199
9-19	ADHCFGU: Audit Management Expert UAP configuration file	200
9-20	ADHSJUAP Audit Management Expert UAP JCL	200
14-1	Sample INSERT statement	321
14-2	Catalog query to extract DBID, PSID, and OBID	322
14-3	DSN1COMP with EXTNDICT parameter	336
14-4	Link-edit dictionary object deck	338
14-5	Linking compressing EDITPROC with dictionary	339

14-6	Preparing the encrypting EDITPROC	340
14-7	Linking the compressing and encrypting EDITPROCs	341
A-1	ftptool	371
A-2	Receive the XMIT files	371
A-3	Data sets allocations	372
A-4	ftp commands	373
A-5	Allocating and tersing	373
A-6	WLM environment	374
A-7	Member SYS1.PROCLIB(WLMENVP)	374
A-8	SYS1.PROCLIB(WLMUTIP)	375
A-9	Customizing GLWRUN	376
A-10	The job BUILD example	378
A-11	Output of DB2 Workload Generator job with action BUILD	379
A-12	Job RUN example	379
A-13	Output of DB2 Workload Generator job with action RUN	380
B-1	DB2 Audit Management Expert for z/OS server configuration file specification	384
B-2	DB2 Audit Management Expert for z/OS agent configuration file specification	389
B-3	DB2 Audit Management Expert for z/OS agent configuration file specification	394

Tables

1-1	PCI DSS requirements	8
3-1	List of RACF resource classes	65
3-2	Cryptography and DB2: options	69
3-3	Example encrypted column VARCHAR size	70
3-4	Description of audit classes	75
3-5	Client accounting strings	78
3-6	Attributes of the segment edit/compression exit routine	81
3-7	IMS log types collected by IMS Audit Management Expert	84
3-8	SMF record type and contexts	85
3-9	SMF Records	89
7-1	Cryptographic hardware per server type	128
7-2	Comparison of System z9 cryptographic hardware	132
9-1	Configuration steps	176
9-2	Installation and configuration sample library members	177
13-1	Maximum record size	314
14-1	SDECSAMP JCL members	318
14-2	Object Identifiers	327
14-3	DB2 treatment of compressed and encrypted data	334
15-1	CKDS verification pattern offsets	358
15-2	Performance measurements for UNLOAD	362
15-3	Performance measurements for LOAD	362
15-4	Performance measurement for REORG	363
15-5	Performance measurement for RUNSTATS	363
15-6	Performance measurement for SQL	363
A-1	GLW table profiles	369
A-2	Data set allocation on z/OS	370
A-3	Driver program parameter - Connection	377
A-4	Driver program parameter - ACTION (BUILD)	377
A-5	Driver program parameter - ACTION (RUN)	378

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.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

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

AIX®	IBM®	System z9®
CICS®	Informix®	System z®
Cognos®	Lotus®	SystemView®
DataPower®	OMEGAMON®	Tivoli®
DB2 Connect™	Optim™	VTAM®
DB2 Universal Database™	OS/390®	WebSphere®
DB2®	RACF®	z/Architecture®
Domino®	Rational®	z/OS®
DRDA®	Redbooks®	z/VM®
DS8000®	Redbooks (logo)  ®	z/VSE™
eServer™	System Storage™	z9®
i5/OS®	System z10™	zSeries®

The following terms are trademarks of other companies:

Cognos, and the Cognos logo are trademarks or registered trademarks of Cognos Incorporated, an IBM Company, in the United States and/or other countries.

Novell, SUSE, the Novell logo, and the N logo are registered trademarks of Novell, Inc. in the United States and other countries.

Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.

Red Hat, and the Shadowman logo are trademarks or registered trademarks of Red Hat, Inc. in the U.S. and other countries.

SAP NetWeaver, SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

J2EE, Java, JDBC, JVM, Solaris, Sun, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Access, Convergence, Excel, Microsoft, SQL Server, Windows Server, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

Preface

In this age of complex regulatory oversight and wide ranging threats to corporate data, securing a company's information assets from internal and external threats has become a primary focus and concern for information professionals. IBM® understands these requirements and using features of the System z® hardware platform, DBMS and operating elements for DB2® on z/OS®, and information management tools can provide a defense that can assist in providing information confidentiality, integrity, and availability.

This IBM Redbooks® publication starts with a description of the data governance requirements, with an emphasis on IBM Data Servers Blueprint, including the IBM Data Server Security Roadmap, and general elements of a complete governance approach. Next, using the elements described in the first section, we position and map the specific elements and requirements of the blueprint-based scenario to IBM portfolio of security solutions.

We then focus on some specific elements and capabilities of DB2 for z/OS and System z platform. These capabilities include elements such as network roles and trusted context, exploitation of network encryption capabilities with SSL and IPsec, and native DBMS Encryption. Included are System z hardware and z/OS operating system elements.

Having laid a solid foundation with the previous components, we then take a deeper look at two specific IBM information management tools solutions.

We build scenarios that demonstrate the use of the IBM Audit Management Expert for DB2 for z/OS. We take an in depth look at the IBM Encryption Tool for DB2 and IMS Databases, including an exploration of the new functionality which provides coexistence with DB2 hardware assisted compression.

The team that wrote this book

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 an information management software Project Leader at the International Technical Support Organization, based in the Silicon Valley Lab. He has authored several Redbooks about DB2 for z/OS and related tools, and has conducted workshops and seminars worldwide. During Paolo's years with IBM, in development and in the field, his work has been mostly related to database systems.

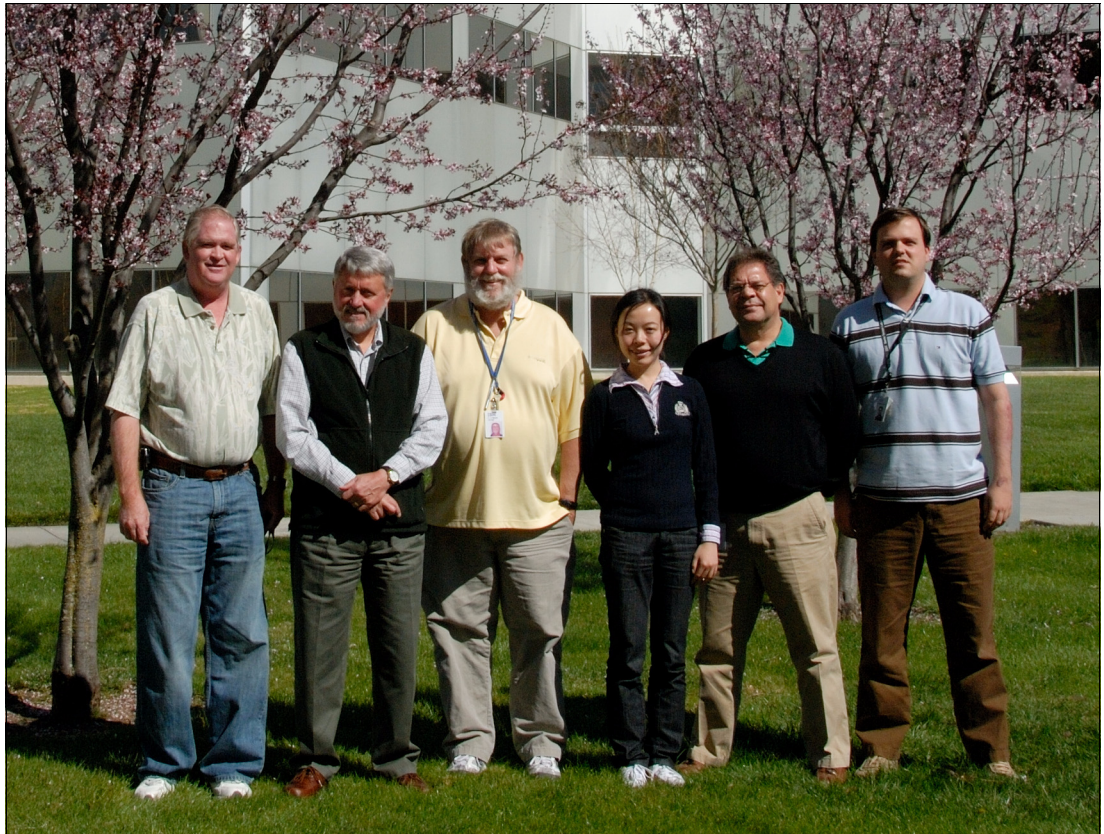
Felipe Bortoletto is a certified IBM IT Specialist in information management. He has 13 years of experience in IT with 8 years of experience with DB2 for z/OS. He joined IBM 4 years ago and is currently a member of the IBM Global Services in Brazil. He holds a degree in Computer Science from UNICAMP

Thomas Hubbard is a Product Specialist Manager for Rocket Software Inc., based in Houston, Texas. He has 28 years of experience in the information technology field. His areas of expertise include both DB2 and IMS backup and recovery planning. He also has extensive experience with system administration and performance management.

Ernest Mancill is a Senior Certified Executive IT Specialist with IBM Software Group. Ernie has 32 years of experience in IT with 17 years of experience with DB2 for z/OS as a Systems Programmer. He joined IBM ten years ago and is currently a member of the IBM SWG DB2 Database Tools technical sales team where he specializes in IBM Information Management Data Governance solutions. His areas of expertise include auditing, encryption, and other data governance solutions on DB2 for z/OS from IBM.

Hennie Mynhardt is a certified IBM Consulting IT Specialist with IBM Software Group. He has lead and worked on various technical projects for database customers in the USA and overseas. His special interests are systems performance tuning and backup/recovery. He currently provides technical consulting and pre- and post-sales support for DB2 for z/OS Engine and Tools.

Shuang Yu is a Software Specialist from IBM China Software Group. She joined IBM China Development Laboratory 3 years ago, working on DB2 Tools for z/OS quality assurance, including Query Monitor, Audit Management Expert for z/OS, Administration Tool and Object Compare. She also performs some technical support for China local customers in DB2 Tools area. She holds a Bachelor's degree in Automation, and a Master's degree in Computer Science from Tossing University.



The authors in SVL. From left to right: Tom, Paolo, Ernest, Shuang, Hendrik, and Felipe

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

Rich Conway
Bob Haimowitz
Emma Jacobs
Mike Schwartz
International Technical Support Organization

Jeff Berger
Jay Bruce
Geoff Jackson
Peter Mandel
Roger Miller
Jim Pickel
Peter Costigan
Tom Vogel
IBM Silicon Valley Lab

Greg Boyd
Tom Hackett
Ernie Nachtigall
System z Security Americas ATS

Marilyn Allmond
System z Crypto and ICSF support

Walid Rjaibi
Belal Tassi
IBM Toronto Lab

Rick Butler
BMO Toronto

Kelly Smith
Rocket Software

Become a published author

Join us for a two- to six-week residency program! Help write a book 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 books to be as helpful as possible. Send us your comments about this book or other IBM 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



Part 1

Data governance

Data governance is the orchestration of people, process, and technology and data to enable an organization to use data as an enterprise asset. Data governance is becoming a regulatory requirement in an increasing number of countries and organizations with possible regular audits.

In this part we introduce the framework for a data governance solution from the data management perspective.

This section contains the following chapters:

- ▶ Chapter 1, “Regulatory compliance” on page 3
- ▶ Chapter 2, “The IBM Data Server security roadmap and some common DB2 for z/OS security themes” on page 19



Regulatory compliance

Today's data governance is a quality control discipline for adding new rigor and discipline to the process of managing, using, improving and protecting organizational information. Effective data governance can enhance the quality, availability and integrity of a company's data by fostering cross-organizational collaboration and structured policy-making.

New advances in IT and in the interdependencies among companies (see GBLA) have made computerized information about business activities a primary source for demonstrating compliance with government regulations. Due to this, regulatory compliance is an increasingly visible, data-intensive, and costly management function that requires collaboration among corporations business units, IT, and its finance functions.

This chapter covers the following topics:

- ▶ "Recent events" on page 4
- ▶ "IBM data governance roadmap" on page 5
- ▶ "Regulations" on page 7

1.1 Recent events

With high-profile data breaches and incidents skyrocketing, the challenge to protect and manage data has become a universal concern for organizations. These are some of the high profile news:

DBA steals personal data, July 2008

A former database administrator at a check services company admitted that he stole and then sold the personal data of about 8.5 million consumers. Court records indicated that this individual stole a variety of personal data from the company's databases over a five-year period starting in February 2002. This information was sold to data brokers through an intermediary for \$580,000.

Laptop, memory sticks stolen in UK government department, July 2008

A UK government agency confirmed laptop containing sensitive information had been stolen while one of their officials checked out of a hotel. A department spokesman said the theft from the hotel in the Liverpool city center on a weeknight brought the total of laptops stolen to 659. The department also said 26 portable memory sticks containing classified information had been either stolen or misplaced since January 2008.

Customer data theft, May 2006

A computer services company that handles most of the credit card, debit card, check, and merchandise return transactions for most of large retail chains across the United States had their computers hacked, putting their shoppers at risk of identity fraud. The breach of credit and debit card data was initially thought to have lasted from May 2006 to January 2007. However, the company said in February 2009 that it now believes those computer systems were first compromised in July 2005. At least 45.7 million credit and debit cards were stolen by hackers.

The theft of millions of customers' credit card information from the services company continues to wreak havoc on this retailing giant. The company announced in July 2007 that it had to absorb a \$118 million charge related to the massive security breach. In April that same year, this same company was hit with a class-action lawsuit seeking tens of millions of dollars.

Investigations into this particular case appear to indicate that the company was not in compliance with the Payment Card Industry (PCI) data security standards established in 2004 by American Express, Discover Financial Services, JCB, MasterCard Worldwide, and Visa International. Reports identified three major areas of vulnerability:

- ▶ Inadequate wireless network security
- ▶ Improper storage of customer data
- ▶ Failure to encrypt customer account data

Card data stolen, 2008

A data breach in 2008 at a payment processor company may have compromised tens of millions of credit and debit card transactions. Such figures may make this incident one of the largest data breaches ever reported. This data breach disclosed by this payment processor company may well displace the January 2007 breach mentioned above as the largest ever involving payment data, with potentially over 100 million cards being compromised.

The company, which processes credit and debit card transaction services, said that unknown intruders had broken into its systems sometime last year and planted malicious software to steal card data carried on the company's networks.

Student obtained passwords, February 2009

A college in Massachusetts discovered that a student changed some of his own course grades by altering exam grades, and broke into the registrar's office. The student has been identified. The college has determined that no other student's course grades have been altered.

The college has determined that the same student obtained passwords and read e-mails and documents of five members of the college community. Based on its own investigation, the college does not believe that any credit card numbers, social security numbers, or similar personal information was compromised.

Computer tape lost, February 2009

A department of information services in a midwest US state announced that a computer record of criminal background checks run on more than 800,000 people over the last 12 years was missing from a storage facility. The state department was informed that a vendor providing secure off-site storage for electronic records and computer files was unable to locate a computer tape in its inventory. The tape contained information from criminal-background checks run on approximately 807,000 people over at least 12 years.

Data breach risks

- ▶ A computer containing sensitive employee information (including 382,000 social security numbers) was stolen from an employee's car.
- ▶ A state department of revenue services reported the theft of a computer containing the names and social security numbers of more than 100,000 state taxpayers in the north east. It is s among more than two dozen state government computers reported missing in the last 14 months.
- ▶ Personal information of 28.6 million veterans and active and reserve members of the armed services is at risk after a computer and computer storage device are stolen from a Veterans Department employee's home.
- ▶ The customer database of a company used by employer health care services to provide prescription medicine by mail was breached. In a twist, the company said it learned of the breach in a letter from someone trying to extort money from the company.
- ▶ An employee of a financial company sold California customer's personal and financial data. The consumer data was sold for as much as \$70,000. The theft and sale of the information included as many as 2 million mortgage applicants. The personal information of the mortgage applicants included Social Security numbers.
- ▶ A storage company for a New York bank lost an unencrypted backup tape containing Social Security numbers and bank account information belonging to as many as hundreds of thousands of Connecticut consumers and personal information of millions more nationwide.

1.2 IBM data governance roadmap

Data is spread across multiple, complex silos that are isolated from each other in many IT organizations. There are redundant copies of data, and the business processes that use the data are also redundant and tangled. There is little cross-organizational collaboration, with few defined governance and stewardship structures, roles and responsibilities.

Today, businesses want to use information for maximum performance and profit. They want to assess the value of data as a balance sheet asset, and they want to calculate risk in all aspects of their operations as a competitive advantage in decision-making.

It is for these reasons that data governance has emerged as a strategic priority for companies of all sizes.

Many companies are learning to examine their data governance practices, and searching for industry benchmarks and common frameworks to ground their approach. The IBM Data Governance Council Maturity Model is a breakthrough initiative designed with input from a council of 55 organizations to build consistency and quality control in governance through proven business technologies, collaborative methods, and best practices.

The IBM Data Governance Council is a group of 50 global companies (including Abbott Labs, American Express, Bank of America, Bank of Tokyo-Mitsubishi UFJ, Ltd, Bell Canada, BMO Financial Group, Citibank, Deutsche Bank, Discover Financial, Kasikornbank, MasterCard, Nordea Bank, Wachovia, Washington Mutual, and the World Bank) that have pioneered best practices around risk assessment and data governance to help the business world take a more disciplined approach how companies handle data

1.2.1 Data Governance Council

Announced in December 2008, the IBM Data Governance Council is exploring the use of Extensible Business Reporting Language (XBRL). This is a software language for describing business terms in financial reports, and risk reporting. The IBM Data Governance Council is seeking input from banks and financial institutions, corporations, vendors, and regulators to create a standards-based approach to risk reporting. XBRL could be used to provide a non-proprietary way of reporting risk. According to the Council, the XBRL Taxonomy of Risk could serve as a fundamental building block to enable interoperability and standard practices in measuring risk worldwide. Such standards could enable central banks to manage databases of loss history and trend analyses that could better inform policy makers and member banks helping to minimize risk and produce better returns.

In 2008, the council met to discuss how businesses will handle the enormous amount and complexity of information generated by organizations and financial markets. The following list is a compilation of their findings, which includes the prediction of five imminent, information-related issues:

- ▶ Data governance will become a regulatory requirement in an increasing number of countries and organizations. In some countries, organizations will have to demonstrate data governance practices to regulators as part of regular audits. This will likely affect the banking and financial services industries first, and will emerge as a growing trend worldwide.
- ▶ The value of data will be treated as an asset on the balance sheet and reported by the chief financial officer, while the quality of data will become a technical reporting metric and key IT performance indicator. New accounting and reporting practices will emerge for measuring and assessing the value of data to help organizations demonstrate how data quality fuels business performance.
- ▶ Calculating risk will be used more pervasively across enterprises for small and large decision-making and will be increasingly automated by information technology. Today, in most organizations, risk calculation is done by a select group of individuals using complicated processes. In the future, risk calculation will be automated, providing greater transparency to examine past exposure, forecast direct and indirect risk, and set aside capital to self-insure and cover risk.
- ▶ The role of the chief information officer (CIO) will change, making this corporate officer responsible for reporting on data quality and risk to the board of directors. The CIO will have the mandate to govern the use of information and report on the quality of the information provided to shareholders.

- ▶ Individual employees will be required to take more responsibility for recognizing problems and participating in the governance process to facilitate greater operational transparency and the identification of risk. They will be aided by operational software that will demonstrate common data governance problems and allow employees to self-govern, sponsor and vote on new policies, provide feedback on existing ones, and participate in dynamic data governance.

Data governance helps organizations govern appropriate use of and access to critical information such as customer information, financial details and unstructured content, measuring and reporting information quality and risk to enhance value, and mitigate exposures. IBM work in this area supports and furthers the company's Information on Demand strategy, announced two years ago, that has delivered results through consistent earnings growth, hundreds of new customer wins, strategic acquisitions, and industry-first software offerings.

1.3 Regulations

Addressing compliance and creating an effective governance strategy has different strategies and implementations depending on the industry. At the forefront are financial institutions. They face many operational challenges and increasing regulatory scrutiny. Hence the birth of regulations within the industry which, if not complied with, can bring can potential reputation damage and fines.

1.3.1 Payment Card Industry Data Security Standard (PCI DSS)

The use of payment cards as a form of currency in exchange for goods and services is the cornerstone of the infrastructure supporting economic growth around the world. In the United States alone, the estimated 641 million credit cards in circulation account for about \$1.5 trillion in consumer spending each year.

However, high-profile data breaches have exposed the vulnerability of payment card processors, point-of-sale vendors, and financial institutions that are not properly securing confidential customer information. Facing increasing risk and financial losses resulting from the misappropriation and misuse of customer information, the payment card industry has taken the initiative. The Payment Card Industry Data Security Standard (PCI DSS) represents the payment card industry's response to these breaches. Merchants and retailers who cannot protect consumer payment card information will be held accountable.

The PCI DSS Council was formed by the major payment card brands (American Express, Discover Financial Services, JCB International, MasterCard Worldwide, and Visa Inc.) to provide a forum in which all stakeholders can provide input into the ongoing development, enhancement, and dissemination of the PCI DSS, PIN Entry Device (PED) Security Requirements, and the Payment Application Data Security Standard (PA-DSS). For more information about the PCI DSS, visit the following Web site:

<https://www.pcisecuritystandards.org/>

You can download the specifications document (a set of comprehensive requirements for enhancing payment account data security) from the Web site after signing the licence agreement.

The PCI DSS is a multifaceted set of regulations that defines requirements for implementing security management policies, procedures, network architecture, software design and other critical protective measures. With the goal of improving the security of electronic payments, the PCI DSS represents a unified industry standard for protecting cardholder data that is stored, transmitted, or processed.

The standard includes 12 requirements across six categories, concentrating on data authentication, access control, audits, and data encryption. To comply, companies that handle payment card information are required to establish stringent security policies, processes, and procedures.

The standard covers a range of issues, such as maintaining a secure network, protecting cardholder information, managing risk, implementing control measures, and monitoring test networks. See Table 1-1.

Table 1-1 PCI DSS requirements

A - Build and maintain a secure network	
1	Install and maintain a firewall configuration to protect cardholder data
2	Do not use vendor-supplied defaults for system passwords and other security parameters
B - Protect cardholder data	
3	Protect stored cardholder data
4	Encrypt transmission of cardholder data across open, public networks
C - Maintain a vulnerability management program	
5	Use and regularly update anti-virus software
6	Develop and maintain secure systems and applications
D - Implement strong access control measures	
7	Restrict access to cardholder data by business need-to-know
8	Assign a unique ID to each person with computer access
9	Restrict physical access to cardholder data
E - Regularly monitor and test networks	
10	Track and monitor all access to network resources and cardholder data
11	Regularly test security systems and processes
F - Maintain an information security policy	
12	Maintain a policy that addresses information security

In this section, we examine the six categories of PCI DSS requirements from the point of view of the solutions of the System z and security-provided functions.

Objective A: Build and maintain a secure network

Objective A has two components:

- ▶ Install and maintain a firewall configuration to protect cardholder data.
- ▶ Do not use vendor defaults for system passwords and other security parameters
Use onfiguration standards (NIST/SANS/CIS).

IBM countermeasure

System z contains many elements that introduce controls and protections that can be viewed as the equivalent of firewall technology. Keep in mind also the tremendous diversity of workload that can be supported by System z, and reap the benefit of this technology. From the beginnings, the forebears of System z, and the z/OS operating system have introduced fundamental architecture that enforces resource isolation and limits memory access across different application processes through logical partitioning. The z/OS Network Policy Agent and z/OS System Health Checker, along with Tivoli® zSecure Admin can provide mechanisms to verify that critical parameters are changed from the well-known vendor-supplied default values. All processes that run on System z are subject to control by the IBM Resource Access® Control Facility (RACF®). RACF administration best practices can be enhanced with the use of the Tivoli zSecure Audit Admin product.

The security features that are inherent with System z and z/OS have been well chronicled, and are viewed as being highly secure. The System z can be considered an excellent place to build a layered defense for a Web-facing DMz (demilitarized zone), especially with the use of System z IFL (Integrated Facility for Linux®). Not to be overlooked, System z has been granted some of the highest levels of security certification, including Common Criteria. The System z implementation of logical partitioning (LPAR) has obtained an EAL 5 rating, the z/OS Operating System provides a EAL rating of 4+, and z/VM® carries an EAL 5 rating.

Objective B: Protect cardholder data

Objective B has two components:

- ▶ Protect cardholder data
 - Key management and key rotation
 - Certificate management
- ▶ Encrypt transmission of cardholder data across open, public networks
 - Certificate management
 - Dumps

IBM countermeasure

All of the elements needed to store and process cardholder data effectively, along with the robust security to protect it, are available within the hardware elements of System z. The z/OS Communications Server, a component of the z/OS operating system, provides System z-hosted applications the ability to communicate across the TCP/IP stack using HyperSockets, which provide high speed capability without exposure through an open or public network. The z/OS Communications Server also provides intrusion detection and protection. Intrusion Detection Services (IDS) evaluates the network stack for attacks that would undermine the integrity of its operation.

The fourth requirement addresses encryption. It states that sensitive data must be encrypted while in storage (data-at-rest) and when cardholder data is transmitted across open public networks. System z provides for strong encryption support. The implementations are dependent on what hardware elements are available on the System z and what type of encryption is needed, generally chosen by business requirement.

To protect cardholder data with encryption, some implementations rely on multiple platforms. These implementations have poor coordination and can be viewed as having little consistency. When having to coordinate encryption support and associated activities across multiple platforms, this becomes a point of vulnerability. System z has capabilities to manage encryption across heterogeneous environments

As mentioned earlier, there is a long-standing heritage of features within the z/OS operating system that can contribute to securing data stored on System z. (Some of these elements include Storage Protection Keys, Cross-Memory Services, enforced workload isolation, z/OS Workload Manager, RACF, and z/OS Communications Server.) Architected into System z memory control mechanisms. In particular with z/VM, as memory is swapped out for use by other applications (part of multi-processing), the memory gets erased. Other platforms which claim to support virtualization through the use of virtual machines cannot make this claim.

DB2 offers a regulatory compliance suite with tools to encrypt, test, audit, and safely archive data for long-term retention. These tools include DB2 Audit Management Expert to provide deep level of auditing capabilities for DB2, and IBM Encryption Tool for IMS and DB2 Databases to help implement enterprise class encryption for data at rest while exploiting the latest in System z encryption hardware technology

The following information pertains to the over-the-network protection by the hardware or within z/OS:

- ▶ DB2 9 features the implementation of SSL, AT-TLS, or IPSEC encryption for sending and receiving data
- ▶ z/OS supports SSL, TLS, AT-TLS, IPSec, OpenSSH, and Open-PGP, plus multiple symmetric and asymmetric encryption methods, including TDES and AES.
- ▶ z/OS Communications Server provides z/OS Intrusion Detection Services to complement network-based IDS. It can detect known and unknown attacks, and can detect problems in real time, providing another layer of network defense.
- ▶ RACF provides the facilities to manage access and disallow untrusted networks and hosts.
- ▶ Offloading IPSec activity to a specialty zIIP processor, supported in z/OS 1.8, accelerates processing in the same way System z offloads Java™ execution to one of its zAAP specialty processors. This improves the price/performance of end-to-end encryption
- ▶ FTP, which is cited as a risky protocol in PCI-DSS requirements, can be protected on System z with IPSec, AT-TLS, or SSL

The Integrated Cryptographic Service Facility (ICSF) is a component of z/OS. CPACF (CP Assist for Cryptographic Function) is a feature available on z9® and z10, which while a non-chargeable option, requires that the hardware CE enable its use. When enabled, CPACF adds cryptographic machine instruction support to every general purpose processor, making routine use of cryptography more transparent. For clear key encryption requests much better performance characteristics can be achieved.

In addition to the CPACF facility, which runs on the general purpose processors, there is an additional hardware element, the CEX2C (Cryptographic Express2 Coprocessor) feature, that can be added for a separate cost. This feature is used to support secure key encryption. All secure key cryptographic work is performed within the hardware boundaries of the CEX2C feature. This provides a completely isolated environment, and at no point in time is any data or cryptographic key exposed to operating system storage. It is also referred to as tamper resistant. In the event of someone gaining access to the CEX2C hardware, upon attempting to remove the element from the hardware cabinet, the registers containing the master key values are zeroized¹, thereby protecting the key from inadvertent or intentional exposure or theft.

In conjunction with the CEX2C, you can also perform third-party digital certificate hosting using PKI Services element of z/OS. This element allows z/OS to enable enterprises to become their own certificate authority. This not only reduces the cost associated with relying on an outside service to provide certificates, but also creates a more secure implementation, as it eliminates a trip out through the public network, which introduces an additional security exposure.

Objective C: Maintain a vulnerability management program

Objective C has two components:

- ▶ Use and regularly update anti-virus software or programs.
- ▶ Develop and maintain secure systems and applications:
 - Latest vendor supplied patches (getting them)
 - Separate development and test environments
 - Review of custom code
 - Change management procedures

IBM countermeasure

Satisfying the fifth requirement to maintain anti-virus protection is somewhat less important or difficult on System z than on other platforms because System z does it as part of its operations. The PCI-DSS requirements note that the mainframe, by its nature, is not vulnerable to viruses to the extent that Intel® platforms are, so it is commonly viewed that for PCI, applications and operating system environments on System z are not subject to exposure through anti-virus.

Requirement six, to secure systems and applications, is covered by SMP/E, z/OS software maintenance and installation tool, LPARs (EAL 5), Storage Protection Keys, and System z isolation of sensitive executables as part of its processing routine.

Optim Data Growth provides the ability to archive relationally intact pieces of inactive data from a customer's operational data store. Optim Test Data Management provides a capability to build right-sized representative test data copies with relational integrity, and in conjunction with Optim Data Privacy, provides the capability to mask all of the PCI-DSS sensitive data elements (PAN - Primary Account Number and associated elements).

On a mainframe, in a well-defined security environment, security operations (RACF administration) are separated from systems administration. This separation of roles and responsibilities is as much a part of business prudence as a technology choice. It is like having the person who signs the checks also has the ability to print them. In most implementations, systems administrators have low levels of authority to effect changes to the security environment, and the security personnel have limited system administration privileges.

¹ Stored data have been erased or overwritten.

Objective D: Implement strong access control measures

Objective D has two components:

- ▶ Restrict access to cardholder data on a business need to know.
- ▶ Assign a unique ID to each person with computer access.
 - Two-factor authentication
 - Root/SCHED
- ▶ Restrict physical access to cardholder data.

IBM countermeasure

Under System z, each piece of work is associated with an identity, usually a RACF-assigned identifier, and subsequent access to resources and facilities are based on permissions granted to that identifier by the RACF administrator. Most facilities that provide points of entry or access to System z resources require an authentication exchange of credentials, which includes a password authentication process. There is no way for an interloper with nefarious intent to bypass these controls.

In the System z world, removable media (such as tape) are protected by access controls, and include tape label checking and verification. There is also a mechanism providing for the processing of *foreign tapes* (tapes created on non System z systems or external System z processors). This mechanism is typically tightly controlled by the RACF administrator and must be implicitly invoked by RACF-authenticated users granted access to this facility.

RACF and Tivoli zSecure provide the protection and the means to monitor and audit access to protected resources by both standard and privileged users. This meets Requirement 7 and 9, which mandates restriction of access to those with a business need. Part of an implementation that demonstrates this requirement would be to limit the number of privileged users to the absolute minimum number needed to operate the System z environment.

Requirement 8 addresses the assignment of a unique ID to each person with computer access. RACF's password management includes rules for password values (enhancing data security) and expiration of passwords. In addition, DB2 and WebSphere® Application Server (host to many J2EE™ applications) can be configured to share a trusted context, making that environment more secure

Objective E: Regularly monitor and test network

Objective E has two components:

- ▶ Track and monitor all access to network resources and cardholder data
 - File-integrity monitoring
- ▶ Regularly test security systems and processes
 - File-integrity monitoring

IBM countermeasure

Monitoring and testing is best done from a point of security or the whole process is less than secure. IBM offers a plethora of monitoring tools like RACF that can generate audit records for both successful and failed access attempts. System HealthChecker (built into z/OS), SMF auditing, Tivoli zSecure, Tivoli Consul Insight Manager, and DB2 Audit Management Expert can interface with enterprise-wide auditing through the Tivoli platform to monitor and test the environment wherever it extends. Of course, the more that activity involving customer information is done on System z, the less complex the compliance with PCI-DSS.

Objective F: Maintain an information security policy

Objective F has one component

- ▶ Maintain a policy that addresses information security for employees and contractors
 - Business continuity plan

IBM countermeasure

System z and its use of RACF provides the foundation that allows the customer to create and enforce a security policy for information security served by z/OS and System z. In addition, this foundation is significantly enhanced by IM Tools for DB2 and IMS, Tivoli, Insight Suite, and z/OS Network Policy Agents. As platform decisions are made, particularly when choosing an infrastructure to host a new application workload, deploying these applications on System z will significantly reduce the security implications of an infrastructure choice.

Equally important is to understand that there are going to be different security policies for different types of information and for different platforms. As enterprises succeed and grow through acquisition and customer base expansion, there will be new sources of data, new challenges in data security, and the desire to use existing security elements already deployed. Without a unified security policy, additional complexity can be introduced. IBM believes that the breadth of security elements imbedded within System z, z/OS, provide an extremely robust and secure foundation, and with additional products to enhance the native operating system and server based security, provide a world class environment for secure data hosting.

1.3.2 Basel II

The Basel II accord is the second of the Basel Accords. The full name is “The International Convergence® of Capital Measurement and Capital Standards: A Revised Framework.” The Basel Accords are recommendations on banking laws and regulations issued by the Basel Committee on Banking Supervision. Basel II, published in 2004, creates an international standard that banking regulators can use when creating regulations about how much capital banks should put aside to guard against all types of financial and operational risks. An international standard such as Basel II is believed to help protect the international financial system from the types of problems that might arise, should a major bank or a series of banks collapse.

In practice, Basel II attempts to accomplish this standard by setting up rigorous risk and capital management requirements. These requirements are designed to ensure that a bank holds capital reserves appropriate to the risk the bank exposes itself to through its lending and investment practices. Generally speaking, these rules mean that the greater risk to which the bank is exposed, the greater the amount of capital the bank needs to hold to safeguard its solvency and overall economic stability.

In January, 2009, the Basel Committee on Banking Supervision announced a package of consultative documents to strengthen the Basel II capital framework. These enhancements will become part of a broader effort the committee has undertaken to strengthen the regulation and supervision of internationally active banks.

One portion of these requirements, which deal specifically with credit risks, was already addressed in the first Basel Accord. The key new requirements of Basel II means that banks must also manage operational risks, which include IT threats and the malicious actions of employees. A successful Basel II implementation requires the ability to take an enterprise-wide view of business events across multiple systems and quickly deliver accurate and verifiable data.

1.3.3 Gramm-Leach-Bliley Act

The Gramm-Leach-Bliley Act (GLBA), also known as the Gramm-Leach-Bliley Financial Services Modernization Act, enacted in 1999, is an act of the United States Congress. The GLBA repealed part of the Glass-Steagall Act of 1933, opening up competition among banks, securities companies, and insurance companies. The Glass-Steagall Act prohibited a bank from offering investment, commercial banking, and insurance services.

GLBA allowed commercial and investment banks to consolidate. An example is Citibank, which merged with Travelers Group, an insurance company. In 1998 it formed the Citigroup, a corporation combining banking and insurance underwriting services under brands. This included Smith-Barney, Shearson, Primerica, and Travelers Insurance Corporation. This combination, announced in 1993 and finalized in 1994, would have violated the Glass-Steagall Act and the Bank Holding Company Act by combining insurance and securities companies. The law was passed to legalize these mergers on a permanent basis. Historically, the combined industry has been known as the financial services industry.

In terms of this compliance, the key rules under the act include The Financial Privacy Rule. The act governs the collection and disclosure of customers' personal financial information by financial institutions. It applies also to companies who receive such information, regardless of whether they are financial institutions. The Safeguards Rule requires all financial institutions to design, implement, and maintain safeguards to protect customer information. The Safeguards Rule applies not only to financial institutions that collect information from their own customers, but also to financial institutions (such as credit reporting agencies) that receive customer information from other financial institutions.

GLBA compliance is mandatory. It does not matter whether a financial institution discloses non-public information or not. There must be a policy in place to protect the information from foreseeable threats in security and data integrity. Under the GLBA, financial institutions must provide their clients a privacy notice that explains what information the company gathers about the client, where this information is shared, and how the company safeguards that information.

1.3.4 Health Insurance Portability and Accountability Act

The Health Insurance Portability and Accountability Act (HIPAA) was enacted by the U.S. Congress in 1996.

According to the Centers for Medicare and Medicaid Services (CMS) Web site, Title I of HIPAA protects health insurance coverage for workers and their families when they change or lose their jobs. Title II of HIPAA, known as the Administrative Simplification (AS) provisions, requires the establishment of national standards for electronic health care transactions and national identifiers for providers, health insurance plans, and employers. It helps people keep their information private.

The AS provisions also address the security and privacy of health data. The standards are meant to improve the efficiency and effectiveness of the nation's health care system by encouraging the widespread use of electronic data interchange in the U.S. health care system.

The HIPAA Privacy Rule regulates the use and disclosure of certain information held by *covered entities* (Generally, this term refers to health care clearinghouses, employer sponsored health plans, health insurers, and medical service providers that engage in certain

transactions.) It establishes regulations for the use and disclosure of Protected Health Information (PHI). PHI is any information held by a covered entity that concerns health status, provision of health care, or payment for health care that can be linked to an individual.

1.3.5 California Security Breach Information Act

The California Security Breach Information Act (SB-1386) is a California law requiring organizations that maintain personal information about individuals to inform those individuals if the security of their information is compromised. This act stipulates that if there is a security breach of a database containing personal data, the responsible organization must notify each individual for whom it maintained information. The act, which went into effect in 2003, was created to help stem the increasing incidence of identity theft.

This law applies to any person or companies that conducts business in California and owns or maintains computerized personal data. Essentially, it covers people's last names and first names or initial, when the names exist in combination with Social Security numbers, drivers' license numbers, or credit card or debit card numbers with passwords. Only unencrypted data falls under the law.

Once a breach has been discovered, the affected company has to notify California residents quickly. Substitute notice is possible if the agency demonstrates that the cost of providing notice would exceed \$250,000, or that the number of customers to be notified exceeds 500,000, or the agency does not have sufficient contact information. Substitute notice consists of e-mail, posting on the agency's Web site, and notification to major statewide media.

1.3.6 Sarbanes-Oxley Act

The Sarbanes-Oxley Act of 2002 is also known as the Public Company Accounting Reform and Investor Protection Act of 2002 and commonly called Sarbanes-Oxley, Sarbox or SOX. It is a United States federal law enacted in response to a number of major corporate and accounting scandals including those affecting Enron, Tyco International, Adelphia, Peregrine Systems, and WorldCom. These scandals caused investors to lose billions of dollars when the share prices of the affected companies collapsed, shaking public confidence in the nation's securities markets.

The legislation establishes new or enhanced standards for all U.S. public company boards, management, and public accounting firms. It does not apply to privately-held companies. The act contains 11 titles, or sections, ranging from additional corporate board responsibilities to criminal penalties, and requires the Securities and Exchange Commission (SEC) to implement rulings on requirements to comply with the new law.

The act establishes a new quasi-public agency, the Public Company Accounting Oversight Board, or PCAOB, which is charged with overseeing, regulating, inspecting, and disciplining accounting firms in their roles as auditors of public companies. The act also covers issues such as auditor independence, corporate governance, internal control assessment, and enhanced financial disclosure.

Sarbanes-Oxley contains 11 titles that describe specific mandates and requirements for financial reporting. Each title consists of several sections, summarized as follows:

- ▶ **Public Company Accounting Oversight Board (PCAOB)**

Title I consists of nine sections and establishes the Public Company Accounting Oversight Board, to provide independent oversight of public accounting firms providing audit services ("auditors"). It also creates a central oversight board tasked with registering auditors, defining the specific processes and procedures for compliance audits, inspecting and policing conduct and quality control, and enforcing compliance with the specific mandates of SOX.

- ▶ **Auditor Independence**

Title II consists of nine sections and establishes standards for external auditor independence, to limit conflicts of interest. It also addresses new auditor approval requirements, audit partner rotation, and auditor reporting requirements. It restricts auditing companies from providing non-audit services (for example, consulting) for the same clients.

- ▶ **Corporate Responsibility**

Title III consists of eight sections and mandates that senior executives take individual responsibility for the accuracy and completeness of corporate financial reports. It defines the interaction of external auditors and corporate audit committees, and specifies the responsibility of corporate officers for the accuracy and validity of corporate financial reports. It enumerates specific limits on the behaviors of corporate officers and describes specific forfeitures of benefits and civil penalties for non-compliance. For example, Section 302 requires that the company's principal officers (typically the chief executive officer and chief financial officer) certify and approve the integrity of their company financial reports quarterly.

- ▶ **Enhanced Financial Disclosures**

Title IV consists of nine sections. It describes enhanced reporting requirements for financial transactions, including off-balance-sheet transactions, pro-forma figures and stock transactions of corporate officers. It requires internal controls for assuring the accuracy of financial reports and disclosures, and mandates both audits and reports on those controls. It also requires timely reporting of material changes in financial condition and specific enhanced reviews by the SEC or its agents of corporate reports.

- ▶ **Analyst Conflicts of Interest**

Title V consists of only one section, which includes measures designed to help restore investor confidence in the reporting of securities analysts. It defines the codes of conduct for securities analysts and requires disclosure of knowable conflicts of interest.

- ▶ **Commission Resources and Authority**

Title VI consists of four sections and defines practices to restore investor confidence in securities analysts. It also defines the SEC's authority to censure or bar securities professionals from practice and defines conditions under which a person can be barred from practicing as a broker, adviser, or dealer.

- ▶ **Studies and Reports**

Title VII consists of five sections and requires the comptroller general and the SEC to perform various studies and report their findings. Studies and reports include the effects of consolidation of public accounting firms, the role of credit rating agencies in the operation of securities markets, securities violations and enforcement actions, and whether investment banks assisted Enron, Global Crossing, and others to manipulate earnings and obfuscate true financial conditions.

- ▶ **Corporate and Criminal Fraud Accountability**

Title VIII consists of seven sections and is also referred to as the Corporate and Criminal Fraud Act of 2002. It describes specific criminal penalties for fraud by manipulation, destruction or alteration of financial records or other interference with investigations, while providing certain protections for whistle-blowers.

- ▶ **White Collar Crime Penalty Enhancement**


Title IX consists of two sections. This section is also called the White Collar Crime Penalty Enhancement Act of 2002. This section increases the criminal penalties associated with white-collar crimes and conspiracies. It recommends stronger sentencing guidelines and specifically adds failure to certify corporate financial reports as a criminal offense.

- ▶ **Corporate Tax Returns**

Title X consists of one section, which states that the chief executive officer should sign the company tax return.

- ▶ **Corporate Fraud Accountability**

Title XI consists of seven sections. Section 1101 recommends a name for this title as Corporate Fraud Accountability Act of 2002. It identifies corporate fraud and records tampering as criminal offenses and joins those offenses to specific penalties. It also revises sentencing guidelines and strengthens their penalties. This enables the SEC to temporarily freeze large or unusual payments.



The IBM Data Server security roadmap and some common DB2 for z/OS security themes

In this chapter, we describe some of the common elements of securing data on IBM data servers and their interpretation for DB2 for z/OS.

The following topics are discussed:

- ▶ “The IBM Data Server Security Blueprint” on page 20
- ▶ “Threat elements of the IBM Data Server Security Blueprint” on page 24
- ▶ “Threat countermeasures” on page 27
- ▶ “Interpretation of some DB2 for z/OS common security themes” on page 34

2.1 The IBM Data Server Security Blueprint

The IBM Data Server Security Blueprint is intended to provide a starting point to help database professionals understand the various avenues of attack that can threaten data stored on DB2 for z/OS, and how the various elements of the System z hardware, z/OS operating system, and DB2 can provide protection to the threats.

2.1.1 Introduction and overview

As discussed in Chapter 1, “Regulatory compliance” on page 3, the topic of data governance covers a broad landscape. Consistent within most of the relevant regulations is the demand for rigorous data custodianship throughout the information life cycle. Numerous initiatives and regulations indicate what needs to be protected, but little or no guidance is given to help organizations define and implement protections. The database professional is placed in an awkward situation, the business demanding compliance and protection, but few specific instructions exist to guide their attempts to conform to these demands.

On hearing this story from customer database practitioners, IBM convened a group of DB2 experts to devise a framework to help customers recognize, classify, and implement countermeasures to protect sensitive customer data. This team, which includes thought leaders in security from the DB2 for LUW, DB2 for z/OS, and Informix® Data Server community, created the IBM Data Server Security Blueprint. The purpose of this document is to assist the customer in the creation of a road map to help in defining and implementing a security plan to protect data on IBM data servers.

The Data Server Security Blueprint is shown in Figure 2-1.

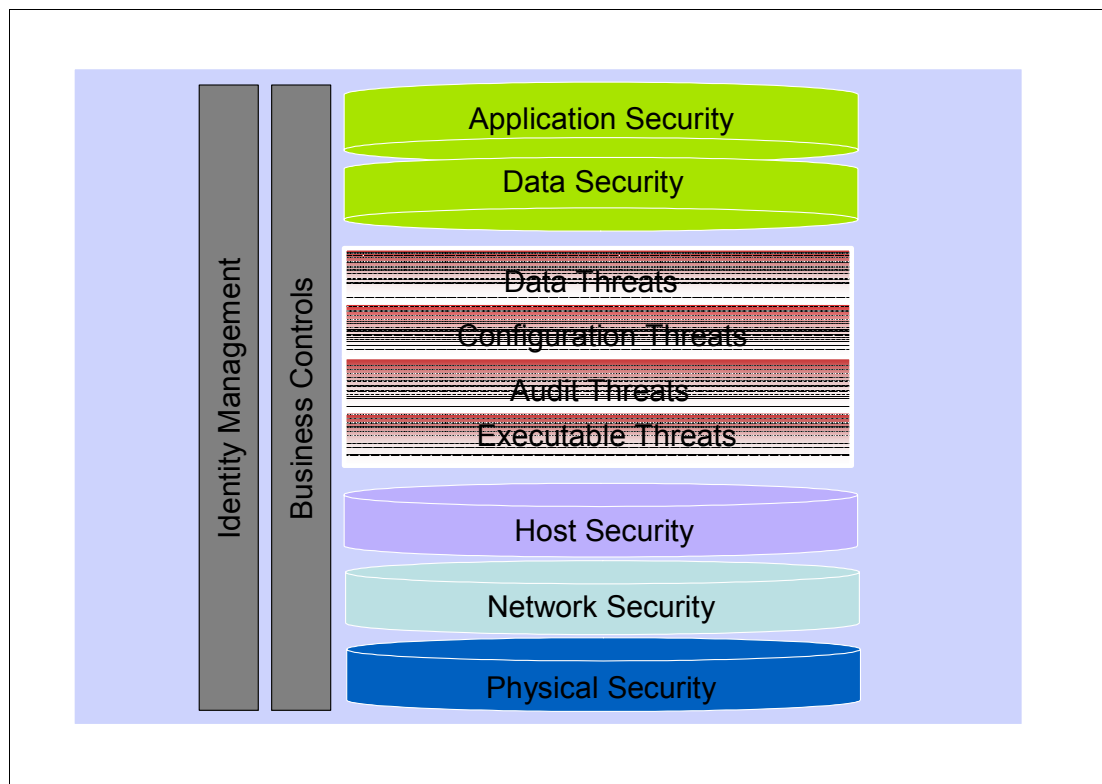


Figure 2-1 The Data Server Security Blueprint

The IBM Data Server Security Blueprint is not intended to be viewed as a definitive implementation plan, but rather to introduce the database professional to a series of categorized threats, avenues of attack, and suggested countermeasures to guard against these exposures. As history proves, when technology continues to advance, there are always new threats and exposures that demand protection, and IBM views the Data Security Server Blueprint as a living document. When new threats and improvements in countermeasure technology surface, the Blueprint document will be updated accordingly.

2.1.2 Why a Data Server Security Blueprint?

Why does the IBM Data Server Security Blueprint exist? As you can probably guess, security and the implementation of a robust security plan is not a trivial task. The IBM Data Server Security Blueprint is designed to help the database professional become conversant in security principals and equip the DBA to participate in the process to build a comprehensive security plan. It is not intended to be a complete or comprehensive checklist for security. Instead, it is designed to help create a framework for data server security, to be supplemented as the unique needs of each customer demands.

The IBM Data Server Security Blueprint helps customers in identifying and understanding most of the common avenues of attack. The blueprint associates these threats with suggested countermeasures. Some of these remedies involve the use of native security features of the DBMS, and others require additional technology. In any event, the countermeasures were suggested by the various teams of platform-specific database security experts and can be viewed as security best practices.

At a minimum, when using the blueprint as your guide and discussing the database perspective for any security plan, make sure that you cover the elements listed in the following sections.

Understand your data

Data and its value mean many things to different people. Before designing your information security framework, you need to answer a couple of questions. The first of these is how to classify your organizations data. To build a complete picture of the landscape of your corporate data and accurately classify each element, you need the input from a number of different groups. Certainly the different line-of-business teams have a good perspective, but other stakeholders exist and should be included in this discussion.

Some points to consider:

- ▶ Does the data element represent information that is considered strategic or proprietary?
- ▶ Does the exposure or theft of the data element carry a legal, monetary, or regulatory cost?
- ▶ What are direct and indirect costs of this data element being lost?
- ▶ Is there an implied or explicit expectation that this data is to be protected on the behalf of your external customers?
- ▶ Would you be upset if it was your data that was stolen?

Data classification can be considered a good first step, and helps identify platforms, DBMS, avenues of access, usage patterns, application relationships, and so on. Understanding these issues can help you determine stakeholders, classify threat avenues, and prioritize countermeasures. Having a complete and representative data model, and access to a tool such as IBM Rational® Data Architect, data modelers or data architects can help the discovery, documentation, and validation of sensitive data. Storing this information inside Rational Data Architect's metadata can demonstrate proper data custodianship when challenged by external auditors.

Need to know

Once you have identified your sensitive data elements, you next need to categorize the users of your data. As with the data discovery exercise described above, this discussion requires the involvement of many different groups (such as line of business (LOB) representatives, security and auditing, human resources, and legal), besides IT development and production support personnel. One approach is to identify specific business roles, categorize data access appropriate to these roles, and associate individuals to these roles. Access to sensitive data elements should not be given to individuals but rather to roles. One overriding rule in this classification process is to permit access to the least amount of data as much as possible. In other words, give each identified role access to only the data absolutely needed to perform that job, and associate individuals to as few roles as possible. There are several advantages to this approach:

- ▶ Justification and defense of access levels are performed at the role rather than at the individual level.
- ▶ As individuals job responsibilities change, role-based access control is more accurate.
- ▶ Role-based privileges lend themselves to implementing DBMS security features such as roles and network trusted context in DB2 for z/OS.

Part of this discussion should be directed at the issue of trusted or privileged user access. Great effort should be taken to restrict the proliferation of these user classifications. While lengthy debate can occur over the need for these super users, it goes without much argument that the number of these users should be limited, controls need to be put in place that ensure robust and frequent password changes occur, and that uncontaminated auditing be conducted to understand whenever actions are performed under the control of these privileged user credentials.

Situational awareness

To coin a phrase with military connotations, this describes an exercise where threat avenues are categorized and identified. The various categories and classifications of threats to your data have been well chronicled in different sources. You need to assess these and decide which ones apply to your specific circumstance. One example of this would be if your organization does not allow the use of USB storage devices, and none of the PC assets contain USB ports, it would make little sense to list this as an avenue of attack.

In classifying and listing threats to your organization and your data, there is a subset of attack vectors to which the database professional has a direct role in implementing the appropriate countermeasure. To help with the identification of these attack avenues, one element of the IBM Data Server Security Blueprint includes a detailed classification of the different types of data threats. Because the authors of the blueprint document were focused on DB2 and IDS, many of the identified threats are of direct relevance to the DB2 for z/OS practitioner. The blueprint should give the DBA their road map and help classify threats and associated countermeasures.

Defense in depth

There are going to be a number of ways that your data could be subject to attempted breach. Many of the necessary countermeasures entail the implementation of technology which is unrelated to DB2 for z/OS. However, without a comprehensive defense plan, the DBA might tightly protect DB2, but weakness in other areas (compromised network security for example), would still leave enterprise data exposed. For this reason, the IBM Data Server Security Blueprint is presented as a small part of a much larger approach. Robust security has to be built in many layers, and the focus of the protection of DB2 for z/OS data assets is but one layer. When implemented, a well-layered security framework ensures that if one layer is somehow compromised, there is a high likelihood that the other well-secured layers would

compensate for this weakness. One example of this would be if a successful attack allowed a user to run an UNLOAD utility against a table with sensitive data. The TCP/IP security policy would protect from the use of FTP to transfer the stolen data outside the protected network.

It is important to understand that protections described in the IBM Data Server Security Blueprint are part of a much larger framework. For a company and its data to be protected, many other layers of protection must be implemented. These other layers comprise countermeasures to protect physical assets, access into and out of the network, and host or server security. While not specifically addressed in the IBM Data Server Security Blueprint, we will discuss some of these security features of System z servers and the z/OS operating system in Part 3, “System z synergy” on page 125.

Shake the locks

Once your organization has gone to the expense and effort to implement security mechanisms; periodic and robust testing needs to be performed to validate that the different security elements are indeed working as desired. As mentioned earlier, the threat environment is constantly changing, and protections deemed secure today might be compromised tomorrow. To truly feel secure when leaving your home for an extended time, you always make sure that locks are turned, alarms are set, and lights are left on. The same concept applies with your data security framework, you must constantly validate that it is truly secure. This validation should encompass not only vulnerability testing (to detect existing gaps in your plan), but also should include penetration testing (to validate that countermeasures, when applied, are providing the expected protection).

Big Brother is watching

One inescapable facet of data security (and in the mind of many database professionals an unappetizing one), is the issue of auditing. Many times this topic is viewed from the context of trustworthiness, that when the actions of database professionals are subject to auditing there is an implication of guilt. This cannot be further from the truth. As privileged users in a secure environment, whenever a data breach occurs, many times the first effort at forensics is aimed at privileged user activities. With a secure and comprehensive auditing framework in place, there can be little doubt cast as to the activities of privileged users in the course of their legitimate administrative activities.

Auditing helps provide a historical perspective of data access. While the primary use of audit data is in the process of performing data breach forensics, there are other benefits with the collection and analysis of this data. Analyzing audit data can help detect patterns of inappropriate access, validate the business rules revolving around need-to-know, and highlight undiscovered avenues of exposure. As we have seen in many circumstances, it can be many months before the detection of a data breach occurs. With regular examination of audit data, early detection of an attack can be achieved, with a corresponding reduction of the ultimate business exposure. Collecting and examining audit data provides concrete feedback to validate the implementation of a data security framework.

2.1.3 Invest in the future

Once you put your security framework in place, you can sit back and rest assured that all is well with the world, right? No. Threats to you and your company's data are always evolving, being constructed in increasingly complex fashions, and arriving with increasing frequency. At the same time, the face of your organization data is changing as well. New applications, exploitation of new technology, merger and acquisition activities, and other business evolutions all bring new security challenges.

Maintenance and improvements to your security framework need to be integrated into all of your information and application methodology. When changes to your data infrastructure are being considered, make sure that the implications to the data security framework elements are understood. Keep abreast of the changing data security landscape, understand as new threats arise, and be aware of improvements in technology that can enhance your security infrastructure. As your organization continues to change and grow, give the data security framework the proper attention to ensure it evolves as well.

2.2 Threat elements of the IBM Data Server Security Blueprint

This section discusses the different categories of threats that are presented in the blueprint, and how they might manifest themselves in the DB2 for z/OS environment.

2.2.1 Data security layers

The IBM Data Server Security Blueprint has identified four major threat categories that apply to data stored within DB2 for z/OS:

- ▶ Data threats
- ▶ Configuration threats
- ▶ Audit threats
- ▶ Executable threats

What do we mean when we speak of a threat? A threat, in the context of the IBM Data Server Security Blueprint, refers to a mechanism by which an action can be initiated with malicious intent, which results in a violation of established security policy. It needs to be understood that these actions are not always executed by unauthorized individuals. In some situations, and specific to certain threats categories, the actions of authorized or trusted users can be contrary to established need-to-know boundaries. The IBM Data Server Security Blueprint further defines these threat categories as follows.

2.2.2 Data threats

Threats against data are mechanisms whereby data can be accessed by users or processes that are not authorized to access such data. This is by far the largest category of threats. These threats can be aimed directly at the tables in the database, or through more indirect means such as by looking at the log files or directly at the table space files on the operating system. These data threats can consist of the following types:

- ▶ Data.1. Connection threat

These are a result of poorly implemented network and connection authentication and authorization. Some common examples include the use of a single, well known authorization ID on a DB2 Connect™ or WebSphere Application Server to authenticate to the DB2 for z/OS server.

- ▶ Data.2. Base tables threat

Poorly implemented security and controls on base tables. For example, granting PUBLIC access to the DB2 catalog and directory tables, with subsequent security concerns about the contents of the various catalog tables.

- ▶ **Data.3. Other tables threat**
 Poor authorization controls on other types of DB2 artifacts. These would include things such as MQTs (Materialized Query Tables), OLAP Cubes, cloned tables, and so forth.
- ▶ **Data.4. Common user ID threat**
 Lack of granular credentials used to connect to DB2 in multi-tiered application environments. This could include the granting of privileges to a single ID that is used to authenticate to the DB2 for z/OS server by many distinct users through the use of a DB2 Connect gateway.
- ▶ **Data.5. Database administration access threat**
 Intentional abuse of privileges inherent in IDs that are given database administration authority. This is known as trusted user access.
- ▶ **Data.6. OS administrator access threat**
 Intentional abuse of privileges by IDs with special server or operating system access. The nefarious actions of the DB2 or z/OS system programmer would fall into this category.
- ▶ **Data.7. Data in transit threat**
 This category of threat includes the exposure of IDs, passwords, and data in the presence of network sniffer technology. These can include diagnostic and performance tools whose use, while legitimate, can intercept sensitive information.
- ▶ **Data.8. Backups**
 Exposure in the form of recovery assets without adequate protections, in particular as these assets are removed from the data center to off site disaster recovery storage locations.
- ▶ **Data.9. Recovery logs**
 Similar to the exposure scenario discussed for backups, transaction and recovery log assets (again used for off site disaster recovery), contain data that needs protection.
- ▶ **Data.10. Archived data**
 The proliferation of operational data has led most organizations to implement archiving solutions to move inactive data to lower tier storage. While no longer considered operational, this data still contains sensitive information and requires the same level of protection as the original source.
- ▶ **Data.11. Diagnostic data**
 This area constitutes data that is gathered to support problem analysis and resolution. This can include assets such as abnormal termination dumps, trace data, SYS1.LOGREC, GTF (Generalized Trace Facility) data, and other sources of diagnostic information. Of particular concern, this class of data is in many cases sent to vendors with little or no downstream control over where the data ends up.
- ▶ **Data.12. Extracted data**
 Many times relational data is unloaded for many purposes, including population of development tables, data to be used as input sources for other types of processing such as CSV format to input into spreadsheet applications, sharing data with off site business partners.

2.2.3 Configuration threats

Configuration threats include threats against configuration mechanisms whereby the database or database manager configuration files can be tampered with. Because they control critical aspects of your data server (such as how and where authentication is performed), it is critical that the database configuration files are protected as securely as the data itself.

- ▶ Config.1.Configuration files

This thread describes how one can make changes to the various configuration parameters that influence how DB2 for z/OS executes. Included in this list would be controls over DSNZPARM, association to specific DB2 authority levels such as SYSOPER, SYSADMIN, or SYSCTL.

- ▶ Config.2.Database creation

Similar to configuration, this addresses ways to influence what types of objects are created and stored in the DB2 catalog. This could include the activities of individuals with authorities such as DBADM, BINDADD, and so forth.

2.2.4 Audit threats

Threats against the audit facility are mechanisms whereby the audit configuration, audit logs, or archive logs can be tampered with. In many cases, audit records are the only way to determine what has happened in the past and the only form of evidence to detect misuse; it is critical that they be able to withstand tampering.

- ▶ Audit.1. Audit configuration

This is similar to the configuration threat mentioned above, but is specific to those elements that control the auditing of DB2 objects and events. This could include the ability to start or stop the DB2 native audit trace with SYSOPER or SYSADM authority, DBADM to alter and remove the AUDIT attribute on tables, and so forth.

- ▶ Audit.2. Audit logs

As audit data is collected, both the source and target destination of the audit data needs to be protected. This protection needs to be especially robust as a clear chain of custody needs to be maintained. Audit data represents the best mechanism for conducting data breach forensics.

2.2.5 Executable threats

Threats against executables are mechanisms whereby database manager executable files can be tampered with. This includes executable spoofing, denial-of-service attacks and Trojan horse attacks.

- ▶ Executable.1. Files

This category of threat would include the load libraries where DB2 applications are stored, the ability to update DBRMs, and the use of BIND. Also of concern is the malicious modification of UDFs and stored procedures.

- ▶ Executable.2. Dynamic SQL

Relatively new, this threat avenue is concerned with the proliferation of tools that rely on the use of dynamic SQL which is then subject to attack through SQL injection.

In the IBM Data Server Security Blueprint model, these threats are identified by a three part name, the category followed by a unique number, and one word identifying the threat. This name takes the following form:

<category>.#.<threat short name>

Throughout the remainder of this book we use this technique to classify the specific category of threat being discussed. For example, when we discuss data encryption as applied to image copy, the threat identifier would be constructed as follows:

Data.8.Backups

Along with these categories of threats and their classifications, the IBM Data Server Security Blueprint also presents recommendations for countermeasures. The remainder of this book details these countermeasures from the DB2 for z/OS perspective. For two of these countermeasures tools, DB2 Audit Management Expert for z/OS and Data Encryption for IMS and DB2 Databases Tool, we take an in-depth look into each solution.

The IBM Data Server Security Blueprint can be located at the following Web page:

<http://www.ibm.com/software/data/db2imstools/solutions/security-blueprint.html>

2.3 Threat countermeasures

For each of the identified threats, the Data Servers Security Blueprint provides a list of countermeasures, many of these are discussed in detail in later sections of this book.

Figure 2-2 on page 28 lists threats and countermeasures of the Data Server Security Blueprint.

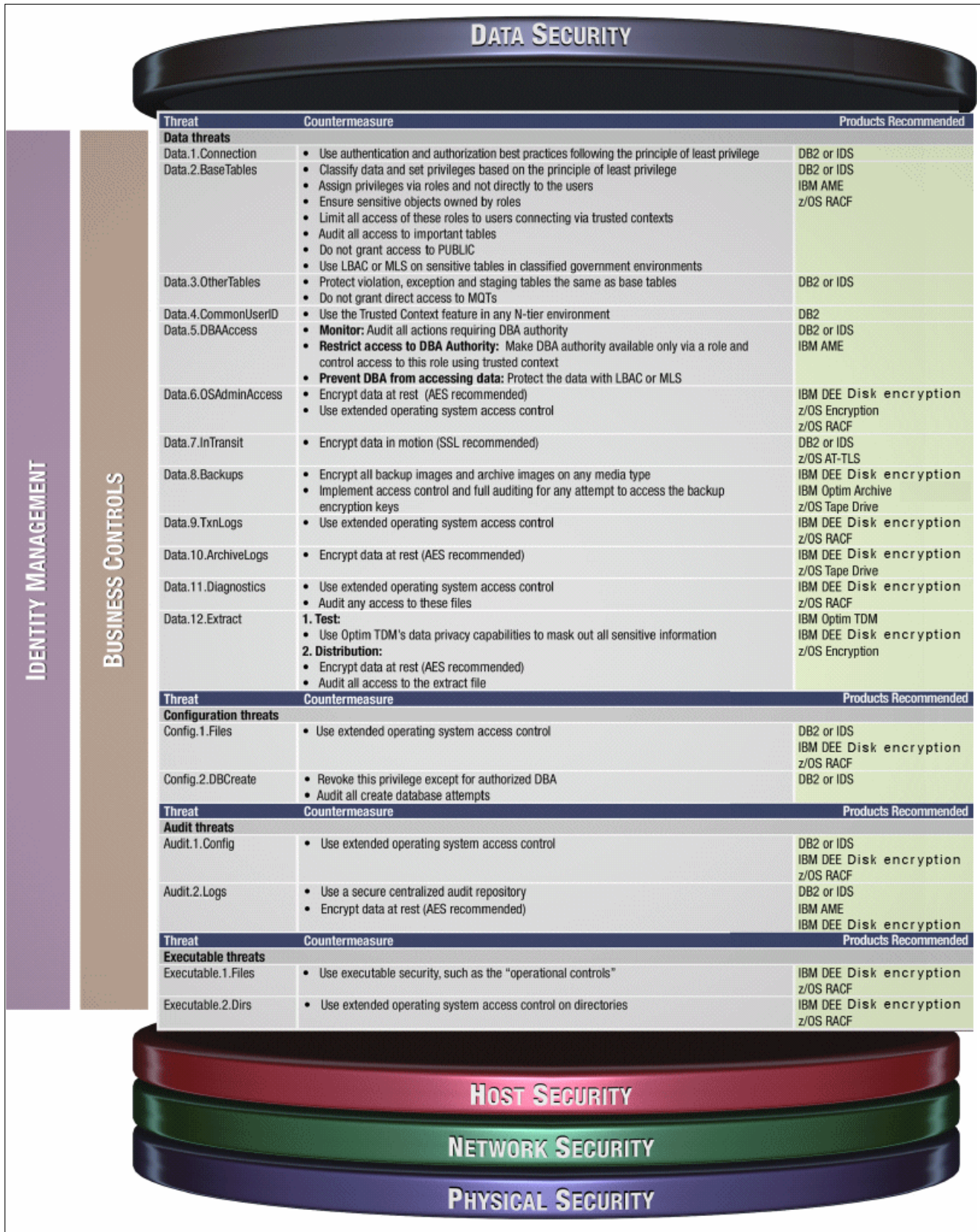


Figure 2-2 The IBM Data Server Security Blueprint

2.3.1 Data threats

In this section we provide a description of the data threats listed in Figure 2-2 on page 28.

Data.1.Connection

This threat is where poor database authentication and connection authentication can allow attack avenues. The protection from this attack vector is to use best practices with connection authorization and authentications. These would include the implementation of Network Trusted Context when defining identifiers and credentials used to establish connections between the application or DB2 Connect gateway server and the DB2 for z/OS data server. A powerful security enhancement introduced with DB2 V9 for z/OS, Network Trusted Context support addresses the problem of establishing a trusted relationship between DB2 and an external entity such as a database administrator or a middleware server. With trusted context support, a series of trusted attributes is evaluated to determine whether a specific context can be trusted. After a trusted context is established, you can define a unique set of interactions between DB2 and the external entity, such as a middleware server, so that the existing database connection can be used by a different user without requiring authentication of the new connection user.

It can establish a connection as trusted when connecting to DB2 for z/OS from a specific location. Having established the connection, it provides the possibility of switching to another user ID, giving the opportunity of taking on the identity of this other user ID only within the trusted context. In addition, it is possible to assign a role to a user of a trusted context.

Ensure that any ID used to establish network connectivity has the minimum level of privileges necessary.

Data.2.BaseTables

This threat concerns poor authorization controls on base tables. Part of the recommended countermeasure entails the use of security best practices as part of the DB2 for z/OS database administrator. Some of these are as follows:

- ▶ Make sure that when explicit privileges are granted, that the least level of privilege is granted. In other words, provide only what is needed for a user to perform their job.
- ▶ Conduct a need-to-know analysis. Collect all users with privileges on a particular object, and revoke privileges that have no supporting access justification.
- ▶ In a DB2 Version 9 for z/OS environment, consider the implementation of role-based security, and grant privileges on objects to roles rather than individuals.
- ▶ Use the PUBLIC privilege when absolutely justified, and ensure that only objects that justify this open level access are allowed. One common technique is for DBAs to grant PUBLIC access to the DB2 catalog and directory objects. This is generally a bad security practice and should be carefully reviewed.
- ▶ For specific situations, consider the implementation of the use of DB2 Multi-Level Security scheme. This allows fine-grained security to be applied at the row level and provides the ability to classify data in different categories.
- ▶ Implement robust auditing controls, such as provided with the DB2 Audit Management Expert for z/OS to understand and track access patterns against base tables.

Data.3.OtherTables

This threat describes an exposure with data stored in other types of DB2 objects, including Materialized Query Tables (MQT), staging tables used for purposes such as replication solutions like DB2 Data Propagator, or OLAP Cubes. In general, the same sets of

countermeasures discussed for base tables apply for this category of data as well. In most cases, these types of objects are for internal use only, so the need to know justification for direct access should be limited.

Data.4.CommonUserID

This describes the threat where credentials are assigned with a single shared ID used to connect application servers and network gateway servers with the DB2 for z/OS data server. The use of the network trusted context implementation would help use the context to perform user authentication. In addition, applications should exploit the use of the SQLESETI API to populate workstation identifiers that can be reflected in the instrumentation facility of DB2 for z/OS. For more information about the use of SQLESETI, refer to the Developeworks article *Monitoring WebSphere Applications on DB2 Servers*, available from the following Web page: <http://www.ibm.com/developerworks/data/library/techarticle/0212shayer/0212shayer.html>

Data.5.DBAAccess

This is the threat where privileged users or DBAs abuse their privileges in an attempt to inappropriately read or modify data. The appropriate countermeasure for this insider abuse requires a layered defensive approach:

- ▶ Understand whenever a privileged user alters resources or accesses data on DB2 for z/OS. Because the inappropriate use of their privileges results in the ability to access the data, the next best thing is to understand the details behind privileged user access through the use of an auditing solution such as DB2 Audit Management Expert for z/OS. Once implemented, whenever privileged user activity occurs, the auditor can challenge the privileged user and ask for justification that governs the use of the access privilege.
- ▶ Associate DBA authorities and only connect the DBA to the role appropriate for the object being maintained. Once the DBA is performing the activities granted to the role, use DB2 Audit Management Expert for z/OS to audit the activities performed under that role. Once the administration task is complete, remove the role from the DBA authorization ID.
- ▶ For extra-sensitive data, or in environments that demand the highest level of security, implement an MLS-based security scheme that restricts access to specific rows, and restrict access to this data through RACF.

Data.6.OSAdminAccess

Similar to abuse of privileges associated with the database administrator, there are similar threat vectors that can be exploited by operating system or storage administrators in a z/OS environment. At a minimum, ensure that all of the physical data sets have RACF data set accesses rule in place. This data set access protection needs to cover components such as the VSAM linear data sets for table space and index spaces, DB2 bootstrap data set, and DB2 active and archive recovery log data sets. Even with such data set level protection in place, there can be access through other avenues, such as through the use of storage administration products such as DFDSSdss. Using such tools, DB2 data can be exposed without the use of an SQL mechanism. To prevent such avenues to read or copy the physical data, encryption of the data at rest, using a robust encryption mechanism such as AES or TDES encryption using the Data Encryption for IMS and DB2 Databases Tool, or ES8000 disk encryption is recommended.

Data.7.InTransit

Whenever sensitive data is transmitted from application requestors through mechanisms such as WebSphere Application Server, or client requests for DB2 data using DB2 Connect, there are attack mechanisms, such as network sniffers, which require protection. For most situations, the use of SSL or AT-TLS is recommended to encrypt the network flows to and from the DB2 for z/OS.

Note: The use of encryption technology for protecting network flows can impact auditing tools that rely on a network appliance approach. This could include solutions such as DataPower®.

Data.8.Backups

Many times great care and focus is given to the protection of assets used widely and for governance. However, once data leaves the physical data center, the security of physical data assets becomes difficult to ensure. Care must be taken to ensure that asset classes such as archive files, DB2 Image Copy backups, DB2 Archive Recovery Logs, and physical sequential unload data sets are protected. The most consistent countermeasure mechanism for these assets is encryption. For physical media stored on tape, the TS1130 tape drive offers high-performance flexible data storage with support for data encryption. When encrypting DB2 for z/OS data using the Data Encryption for IMS and DB2 Databases Tool, image copy data sets and recovery log records are encrypted by default. For physical sequential data sets, such as unload SYSREC data, the z/OS Encryption Facility provides an encryption mechanism. For other forms of data stored on disk, the ES8000 encrypting disk solution can provide encryption support. If the requirement exists for long term retention of data, with clear demonstration of immutability, the Optim Data Growth Solution can store compressed archived data in its proprietary format. In addition, archive targets can be included using the DR550 tape drive technology, which can incorporate both encrypting tape and a WORM (write once, read many) implementation.

Data.9.TxnLogs

As mentioned, the DB2 recovery log, both archive and active logs, contain potentially sensitive data which requires protection. These data sets are particularly critical as without them, it is difficult to ensure the accurate and timely recovery of the DB2 subsystem and application objects in the event of a failure. At a minimum, robust RACF data set access rules need to be put in place, and encryption should be considered, either through the use of encrypting tape drive or disk technology (such as ES8000 encrypting disks) which provides explicit encryption of the entire log data sets, or through the use of the Data Encryption for IMS and DB2 Databases Tool which implicitly encrypts log records associated with updates against encrypted objects.

Data.10.ArchiveLogs

The same avenue of threat and the same countermeasures would apply to the archive log situation. In addition, when these archive logs are retained for extended periods of time, attention must be given to the matter of key retention and impact of key rotation as it affects encrypted archived data and logs. We discuss the ramifications of this during our encryption scenarios using Data Encryption for IMS and DB2 Databases Tool in Chapter 10, “Audit Management Expert scenarios” on page 211.

Data.11.Diagnostics

This threat is becoming an area of focus for many customers, as the shipment of diagnostic information such as system or console dumps, GTF trace, SMF trace, and other problem determination assets demand secure transportation and protection. Whenever sensitive assets are transported on physical media or through network-based transport mechanisms such as FTP, encryption through Secure Sockets Layer (SSL) provides the best protection. In addition, ensure that the data is stored at the receiving end in an encrypted format. IBM service provides the customer with a service aid utility to prepare diagnostic materials prior to shipping to IBM.

The IBM z/OS Problem Documentation Upload Utility is a parallel FTP utility that is designed to send documentation in a more efficient manner to IBM FTP sites. This utility sections the

input file into smaller files that are sent in parallel, resulting in shorter transmission time for large data sets (such as stand-alone dumps). The parallel FTP program always compresses the input data before it is written to the work data sets. Therefore, it is not necessary to use a tool such as AMATERSE or TRSMAN to compress the input data set before using the parallel FTP program to send it to the IBM FTP site. In addition, 192-bit triple Data Encryption Standard (DES) can be requested by using the CIPHER_KEY keyword. Without the keyword, the data is just compressed. With the keyword, the data is compressed, and then encrypted.

Encryption is provided by the CP Assist for Cryptographic Functions, DES/TDES Enablement (CPACF, feature 3863), and is available on all processors starting with the z990 (2084) and z890 (2086). CPACF, feature 3863 enables clear key DES and TDES instructions on all supported CPs. The encryption algorithm used for TDES is documented in *z/Architecture@ Principles of Operation*, SA22-7832. For more information about the IBM z/OS Problem Determination Upload Utility, or to download a copy, refer to the following Web page:

<http://www14.software.ibm.com/webapp/set2/sas/f/z aids/pd uf.html>

Data.12.Extract

Once data is extracted and moved out of its secure location inside the data center, through either FTP or sequential file on tape or disk, it is subject to attack exploitation. Examination of the purpose for the data extraction results in one of several countermeasures. If the data is being used to populate development environments with test case data, then the Optim Test Data Management solution in conjunction with the Optim Data Privacy solution can provide protection. Using these two solutions, right-sized representative sampling of source data can result in relationally intact subsets of extracted test data. Once extracted, the subset of production data can then be transformed in numerous different ways. The net effect is to obfuscate the original sensitive data in meaningful ways. The result is intact, representative, and realistic test data, with no actual production data exposed.

If, on the other hand, the extract is used for other purposes, such as business partner sharing of unloaded or extracted DB2 data, consumed in its sequential form for non relational processing, or other purposes, then encryption is recommended prior to the data being sent off site. For sequential tape, either encrypting tape technology like the TS1130 tapes, disk encrypting technology such as ES8000 encrypting disks, or the use of the z/OS Encryption facility might be appropriate. For data shared through FTP, ensure that SSL or AT-TLS encryption be used during the FTP transportation process.

2.3.2 Configuration treats

In this section we provide a description of the configuration threats listed in Figure 2-2 on page 28.

Config.1.Files

Poor security or improper access privileges on configuration resources such as DSNZPARMs, DB2 SMPE Target Libraries, and especially the DB2 for z/OS APF Authorized library SDSNEXIT can represent significant levels of threats. Rigorous RACF data set protection needs to be applied to all the listed data sets. In addition, particular attention needs to be paid to the contents of SDSNEXIT. Security needs to include an understanding of who has access to the z/OS operating system privileges that allow for dynamic additions to the APF list. Use of certain z/OS performance monitors allow mechanisms to add members to the APF list dynamically. The use of these facilities also needs to be properly controlled, always audited, and periodically reviewed for justification. DB2 also provides for a mechanism to change many DSNZPARMs dynamically. Control of this mechanism should be put in place, and access to these types of facilities should be strictly limited to the smallest possible audience.

SMPE provides an excellent vehicle to place controls on configuration and software changes in the z/OS, subsystem, and program product environment. Where possible, encourage the use of SMPE to manage all software upgrades performed in the z/OS environment.

Config.2.DBCreate

Limiting the number of IDs which are associated with the special data privileges of DBADM can reduce the potential for unauthorized creation of DB2 for z/OS objects such as databases, tables, or indexes. Examine the number of IDs with these special levels of authorization and revoke any unnecessary or unauthorized ID with this capability. Always audit whenever create attempts are executed; the use of the DB2 Audit Management Expert for z/OS tool provides specific auditing collection of these events.

2.3.3 Audit threats

In this section we provide a description of the audit threats listed in Figure 2-2 on page 28.

Audit.1.Config

When relying on the native audit trace facility of DB2 for z/OS, there are usually a number of IDs that contain privileges that can allow for the interference with the collection of audit trace events. As with other special database privileges, inventory the audience of IDs with the ability to start or stop audit traces, and limit the use of this privilege. The use of an auditing product (such as DB2 Audit Management Expert for z/OS) can ensure clear separation of the control in the collection and operation of the audit environment is demonstrated. The configuration and collection parameters used by DB2 Audit Management Expert for z/OS can be administered with no special DB2 privileges needed.

Audit.2.Logs

With the collection and storing of the audited data, measures need to be taken to ensure the validity and immutable nature of the audit data. The use of a centralized audit data repository managed by a product such as DB2 Audit Management Expert for z/OS can provide a secured audit data vault. If the audit technique relies on the native trace facility of DB2 for z/OS, ensure that the output audit data logs (SMF) are properly protected with RACF data set access controls. For many interpretations, the details contained in the audit data repository and logs are themselves sensitive data requiring extra levels of protection. Consider the use of encryption technology such as the Data Encryption for IMS and DB2 Databases Tool for audit data residing on DB2 for z/OS tables, or the use of the z/OS Encryption Facility or TS1130 encrypting tape drives for SMF audit log data.

2.3.4 Executable threats

In this section we provide a description of the executable threats listed in Figure 2-2 on page 28.

Executable.1.Files

The malicious modification of executables for the most part is a well-protected avenue in most mature z/OS installations. The proliferation of change management software, in conjunction with well-defined RACF data set access rules, make this an unlikely avenue of attack. One recent form of attack, related closely to the executable file attack vector, is that of SQL injection. This takes the form of applications issuing dynamic SQL, and, through various mechanisms, additional SQL statements are entered from the client application. The result is that imbedded within the original SQL statements are additional valid SQL statements with the potential for malicious intent. One effective countermeasure for this type of attack is the

use of pureQuery to examine each of the dynamic SQL statements as they arrive at the DB2 for z/OS server. The dynamic SQL can be replaced, based on prior arrival patterns and application knowledge, with statically bound packages. This provides not only protection from SQL injection, but by substitution of well-tuned static statements, a more consistent and better performing SQL workload would also result.

Executable.2.Dirs

While not directly applicable to the z/OS environment, this concept could be compared to the controls needed to secure the ability to add libraries to the DB2 STEPLIB concatenation: RACF data set access controls limiting the number of IDs authorized to add modules to the STEPLIB, and limitations on access within the APF environment to the fewest number of IDs possible. Another form of attack involves the introduction of nefarious stored procedures, so the extension of the WLM and stored procedure execution environment needs to be controlled, again by the use of appropriate RACF protection services.

2.4 Interpretation of some DB2 for z/OS common security themes

In many situations, there are some topics which need to be discussed within an organization with different viewpoints or agendas. Many of these discussions occur between DB2 database professionals and security and compliance groups. The first part of this section outline some of these themes.

2.4.1 Separation of roles

One consideration in auditing is that any mechanism used to audit activities of trusted users must be implemented in such a way as to prevent the privileged user from interfering with the collection or contaminating the source of the audit data. An auditing solution for DB2, such as Audit Management Expert for z/OS, must maintain the segregation of duties, demonstrate to outside review the guarantee of audit data integrity, which will result in accurate and reflective auditing reports. This separation of roles allows the DBAs to perform their job duties and allows auditors to run audit reports independently of the DBAs, resulting in easier, more accurate audits. Auditors now have the ability to adhere to published industry standards and external auditing without relying on the involvement of personnel being monitored.

A robust auditing solution on DB2 for z/OS must be well-suited to enforce controls that govern DBAs and to report on their activity. DBAs are trusted with sensitive data to do their jobs. They need to maintain, copy, and recover sensitive data, and load and reorganize it, to name a few of their responsibilities. Continuous, automated, and isolated auditing removes the opportunity to alter or even omit important data from the audit reports. Given the potential for DBAs to use their authorization to disable the collection of trace records from the DB2 IFI, an independent audit collection mechanism, which removes the personal involvement of the DBAs, can provide assurance that reported data has not been modified. Consequently, the accuracy of data and reports is more reliable.

The Problem: The DBA perspective

In many cases, DBAs tend to be the focus of audit forensics on DB2 for z/OS, and yet they are involved in the control and collection of audit data used for analysis of their activities:

- ▶ **Audit data collection**

Existing developer and DBA tools are not audit-oriented, nor are they designed to collect all the relevant audit information from the source.

- ▶ Reporting

Existing developer and DBA tools are not audit-oriented, and they are not designed to present information in a useful way for an audit.

- ▶ Integrity

DBAs are part of the audited population, and therefore should not be relied upon to provide key audit information. Furthermore, DBA user identifications (user IDs) have more system-level privileges than typical business users, providing more opportunity to circumvent normal business controls.

- ▶ Workload

DBAs are busy enough tasked with their existing mission to support application development and the normal care and feeding of the DB2 for z/OS environment. Frequent requests for generating audit data and reports can impact DBA productivity, and conflicts in workload can result in delay in the timely generation of audit details.

The Problem: Auditor perspective

Using the native facilities of DB2 for z/OS auditing, auditors usually have to rely on more technical DB2 professionals to create and operate in the native auditing environment, as these require special privileges and a detailed understanding of the DB2 for z/OS environment:

- ▶ Privileges

Auditors generally are not granted the system privileges needed to collect the needed information themselves.

- ▶ IT skills

Even if auditors were given system privileges, they need to develop substantial knowledge of DB2 for z/OS to collect and correlate audit trace data at an application level. Developing such skill is costly, time-consuming, and tangential to the auditor's primary role.

- ▶ Complexity

Because data can be proliferated across the enterprise, it is increasingly difficult to pull information together from all systems.

- ▶ Cost

A more comprehensive audit results in a higher labor cost.

- ▶ Repercussion

A less comprehensive audit runs the risk of missing important events and could allow a company to operate out of compliance.

The DBA and separation of roles

The amount of DBA involvement in the auditing process varies widely. Because different industries are held to different security standards, some audit requirements can result in a substantially greater workload for both the DBA and the audit team. The workload can also be affected by the auditing process itself. For obvious reasons, the less work that is required by the DBA for the auditing process, the better for both the DBA and the auditor. However, while assisting the auditor, the DBA is aware that they too are within the scope of the audit.

Most employees in a company have predefined data access privileges associated with their job role. Prior to the enhanced auditing regulations, it was an accepted practice to allow DBAs and system administrators access privileges to all data. Today, access to sensitive data is split among the DBA and system administrators. While each still has access to sensitive data, they do not have access to data that is not within their business scope. That is, their access to

sensitive data is mostly compartmentalized and each only has the appropriate access to perform their job duties. Despite how DBAs access to sensitive data has changed with the implementation of each regulation, a DBA role in the audit process is still viewed as problematic in most interpretations of the current regulatory compliance landscape.

One other area of interest is the retention of audit data assets, elements such as DB2 audit trace records or DB2 Archive Log data sets for long term forensic analysis. Some customers believe that the simple retention of this “raw” audit data can demonstrate compliance. However, the absence of easy to use reporting mechanisms makes this saved data essentially unusable; few auditors possess the technical expertise to generate meaningful audit reports from this data, and this certainly blurs the line separating the auditor and the DBA from a roles and responsibility standpoint.

The key to gathering data with integrity, meaningful representations of the data, and maintaining a separation of the roles of auditor and DBA, is to automate the process with auditing software. Auditing software gives the auditors independence so they can adhere to published industry standards without relying on personnel who are also being monitored. The right software can help organizations audit more successfully, and less expensively, by providing an easy-to-use tool to access the required data.

From a high-level perspective, the response is usually surprise and dismay at the cost of obtaining data during an audit, which in turn creates the motivation to reduce the cost. A company may wonder if it is necessary to spend money to train auditors to be nearly as experienced as database administrators when the auditor still requires the DBAs help to get the data. It all boils down to a conflict of duties between the auditor and the database administrator.

Certain concerns arise from the company's perspective. Not only are the efforts mostly manual, but how thorough can the audit be using these methods? Was something critical missed? If so, you could end up in reactive mode. The audit data is gathered after the event and could be difficult to find or unavailable.

DB2 Audit Management Expert for z/OS allows for clearly demonstrated separation of roles and responsibilities between auditors and privileged users such as DBAs. Auditors can maintain complete control over the collection of audit data on DB2 for z/OS without requiring special privileges. We take a close look at DB2 Audit Management Expert for z/OS in Part 4, “DB2 Audit Management Expert” on page 161.

2.4.2 Audit versus external security

For some, the fact that the DB2 for z/OS resides in a well-protected RACF environment constitutes sufficient control and auditing is not required. To summarize the relevant points:

- ▶ RACF, and its primary competitors, CA-TopSecret and CA-ACF2, are robust security products for z/OS and all perform an excellent job in protecting access to secured assets on DB2 for z/OS. However, they do little in the way of access and activity reporting. DB2 Audit Management Expert for z/OS is a robust tool that can collect and report on activity performed in the DB2 for z/OS with relatively low overhead. It does not perform or enforce any security policies.
- ▶ In any well-protected environment, there exists a requirement for users to possess special privileges, and the nefarious use of those privileges can allow for the access to sensitive data outside the well-protected application environment. However, the access and use of these privileges are required by specific classes of users, such as database administrators, to provide the smooth operation and administration of the DB2 for z/OS environment.

- ▶ Understanding how trusted (privileged) users access sensitive information is essential to ensuring that data is indeed protected. Robust auditing on the activities performed by these privileged users and the use of certain tools and SQL generation products is clearly needed. In addition, there must be documentation that the source of this audit data is clearly untainted.
- ▶ DB2 Audit Management Expert for z/OS provides the DB2 for z/OS customer an industrial strength auditing tool, with the ability to collect granular levels of activity with relatively low overhead. Organizations can demonstrate a clear separation of roles between the activities of the DBAs and the collection of audit data.

Protect or audit?

Let us begin by describing why, in an environment well-protected by RACF, the requirement for robust auditing still exists. To begin, we need to define a couple of items. For the purposes of this discussion RACF, and its primary competitors CA-TopSecret and CA-ACF2, are security products that provides access control and security functionality for the z/OS and z/VM operating systems. Included in these security products are interfaces that allow for the protection of DB2 for z/OS resources.

While there are some limited reporting capabilities within these security products, they are mainly limited to helping the security administrator with the task of maintaining the security environment, and are of limited value in forensic auditing.

Auditing is about ensuring that the appropriate controls are in place to identify inappropriate access and use of production data. The key point to remember is that auditing solutions do nothing to protect access to data or other DB2 resources.

Customers have historically been averse to performing auditing activities within the DB2 for z/OS environment for several reasons. These reasons include the performance impact of auditing and increased complexity in the management and operation of the DB2 environment. Compounding this issue is the sheer magnitude of audit information that can be collected when not properly filtered. Auditors soon become awash in a sea of audit data. Some customers believe that the simple retention of this raw audit data can demonstrate compliance. However, the absence of easy to use reporting mechanisms makes this saved data essentially unusable. Few auditors possess the technical expertise to generate meaningful audit reports from this data. To simplify the process of turning the sometimes massive amounts of audit data into meaningful and manageable information requires the introduction of easy to use audit management tools, such as the IBM Audit Management Expert for DB2 for z/OS.

The privileged user scenario

To provide the continued health and well-being of any DBMS system, including DB2 and IMS on z/OS, there are many activities that are required to be performed on a regular basis by system and database administrators. These activities, while capable of being controlled by external security processes, such as RACF, are pervasive in effect, and can be used in ways that are contrary to security policies.

To site one possible scenario, assume there is sensitive data residing on a DB2 table, and the applications (CICS® or IMS) that access this table are protected by RACF. The DBA does not have RACF authority to execute the CICS application, but has DBADM authority to administer the table, and in many cases this includes SELECT authority. The DBA runs an UNLOAD utility against the table, extracting all the data contained in the table and transfers that data through any number of mechanisms to an outside entity (FTP, Flash/USB, CSV to spreadsheet, and so forth). Because the user has special privileges against the table, there is no evidence of a security violation as would be reported by RACF.

If, on the other hand, the environment was protected by the use of an auditing solution such as DB2 Audit Management Expert for z/OS, there could be several different collection profiles in effect that would report on this authorized but questionable use of special privileges. One recommendation for audit collection is to monitor any SQL or utility access for privileged users. Conversely one could elect to monitor each utility event, and one could combine looking for one or both classes of events with a time interval. So, while it might be acceptable for the DBA to access the audited tables during normal business hours, auditing parameters might be set up to look for unusual access patterns outside of normal business hours.

The conundrum in all of this is that while these activities and their user permissions can be controlled, the granting of these authorities give the privileged user capabilities to access DB2 and IMS resources and data by means outside the use of the application environment. This has the affect of providing carte blanche access to the data, and to a large extent circumventing normal transaction-level RACF protection.

Auditing the privileged user

As discussed in the prior section, the privileged user needs special authorities and access to DB2 and IMS resources to administer the DB2 and IMS to perform their job. But, in the absence of a robust auditing mechanism to monitor the use of special privileges and data access patterns performed by privileged users, it is impossible to trace when or if these special privileges have been abused. So, in a DBMS environment where privileged user authorities have been granted, there must be some mechanism to track and record activities that are performed under the control of these privileged user identifiers. Audit Management Expert for DB2 and IMS are two solutions that help customers meet this requirement for robust auditing of DB2 and IMS activities.

2.4.3 Personally identifying information and index encryption

Many popular security and compliance initiatives have the requirement to protect specified data elements. The premise is that when exposed, these elements can constitute avenues of financial exposure through the fraudulent use of this information. Some examples of this type of information can include items such as Social Security numbers (SSN), cardholder numbers (also referred to as primary account numbers, or PAN), and driver license numbers.

In many instances, the compliance initiative requires that anywhere these elements are stored in the database instance, they must be encrypted. In some interpretations, this would include not only the information as it exists in the base table, but would also demand encryption on any derived data.

There are several different encryption solutions that are introduced and discussed in Part 5, “Data Encryption for IMS and DB2 Databases Tool” on page 275, but for the DB2 for z/OS customer, there are basically two encryption options available.

The first option, the built-in encryption, initially delivered with DB2 for z/OS Version 8, encrypts at the column level. When columns selected for encryption are also named as index columns, the associated index column value is also encrypted. While protecting the index column value, this approach implies some performance issues:

- ▶ Predicates that depend on sequence of encrypted columns can generate incorrect results
- ▶ Range predicates can produce results with a significant increased performance cost
- ▶ Some statements result in index scans to materialize results

Because of these issues, and other operational and application programming considerations, the use of the built in function for DB2 for z/OS is not wide spread into an existing application environment.

The second IBM encryption solution available to the DB2 for z/OS customer is the Data Encryption for IMS and DB2 Databases Tool. This tool provides implementation without application programming changes, as it implements encryption through the use of an EDITPROC. However, due to the order of the EDITPROC being driven after the index insert/update, while the base table row is encrypted, the associated index rows are left in clear-text. Some interpret this implementation as leaving sensitive data exposed.

Personally Identifying Information (PII) is a concept that describes any data that can be used to establish individual credentials or bona fides. The primary example of this would be information such as name, but could be expanded to include other elements such as birth date, or popular password challenge elements such as mother’s maiden name. When looking at an index design, the improper combination of index columns can create an exposed index. By combining sensitive columns (for example a customer PAN), along with PII data such as customer name, the resulting index row constitutes an exposure. The clear-text PII presented in conjunction with the PAN allows for the nefarious use of the exposed data. In Figure 2-3, the top figure (in red) shows an example of an ‘exposed’ index. As illustrated, even though the data in the underlying base table row is encrypted, the columns in the index are not. In addition, the combination of the PAN, along with the PII columns (in this example first and last name), constitute a situation where one could construct credentials from the index value.

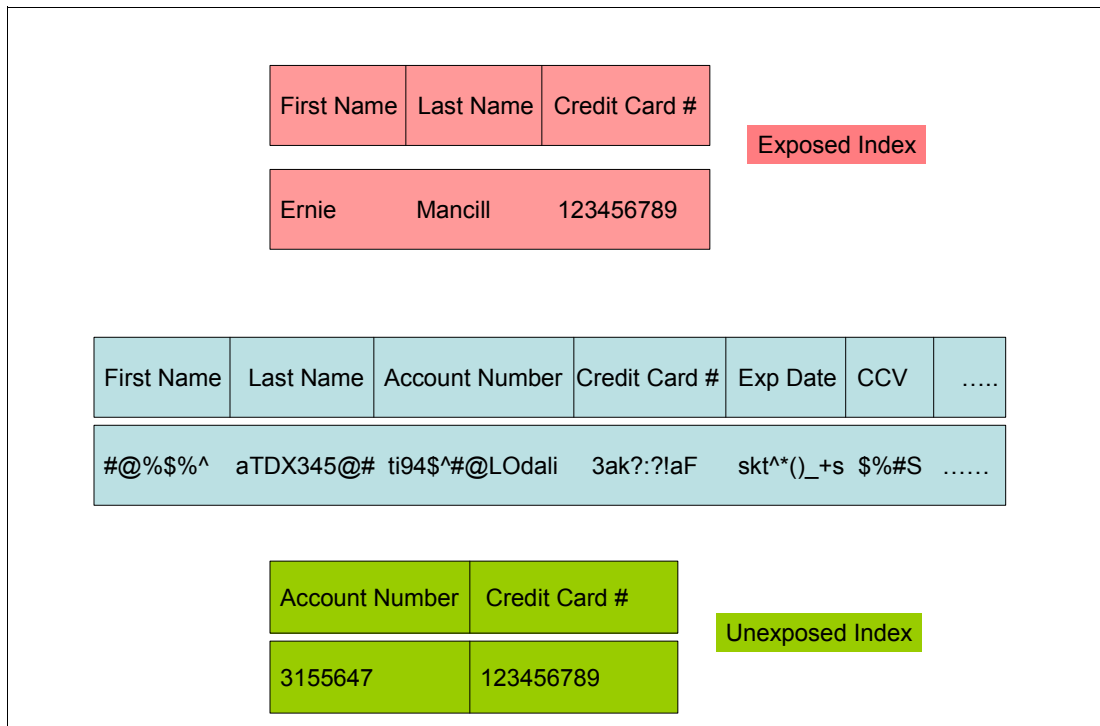


Figure 2-3 Exposed index representation

To use stolen information, such as cardholder account number, the presentation of this information needs to be made in conjunction with one or more pieces of PII. The personally identifying information is validated by some backend process before transaction authorization occurs. The algorithms used to generate sensitive information are well known, so the ability to construct a valid looking credit card or SSN is relatively easy. To use these elements, they need to be presented along with matching credentials which include PII.

A more secure design would consist of an index where sensitive data columns are not combined with columns deemed to represent PII. Just because an index exposes some piece of sensitive data, PAN for example, does not by itself constitute a concern. With such a

design, with the index in the clear, but the underlying base table encrypted, and the PII protected, there is no avenue to use the PAN because there is no associated PII that can be used when challenged to establish a relationship between the stolen PAN and credentials such as a valid name issued to use the stolen PAN. This type of index can be described as an unexposed index. The second index, represented as the bottom diagram (in green) on Figure 2-3 on page 39, demonstrates this form of index.

If, on analysis of a particular index, it has been determined that the index exposes a combination of PII and sensitive data, there is an approach that can be used to limit the amount of exposure. One can introduce the use of another table, with an associated index, which would allow for separation between the PII and sensitive information in the exposed index. The PII in the original index would be replaced with some form of programmatically generated reference number, which would also be stored in the base table. The PII would then be moved from the original base table into a new reference table. The application would then access the original table, using the sensitive information (PAN) as key. The application would then extract the reference number and retrieve the PII using the reference number as key. While not transparent, this would provide separation of PII and PAN, remove their exposure together in a single index, and would provide for encryption of the underlying tables.

In looking at Figure 2-4, the first diagram at the top of the figure represents an unacceptable or exposed index access.

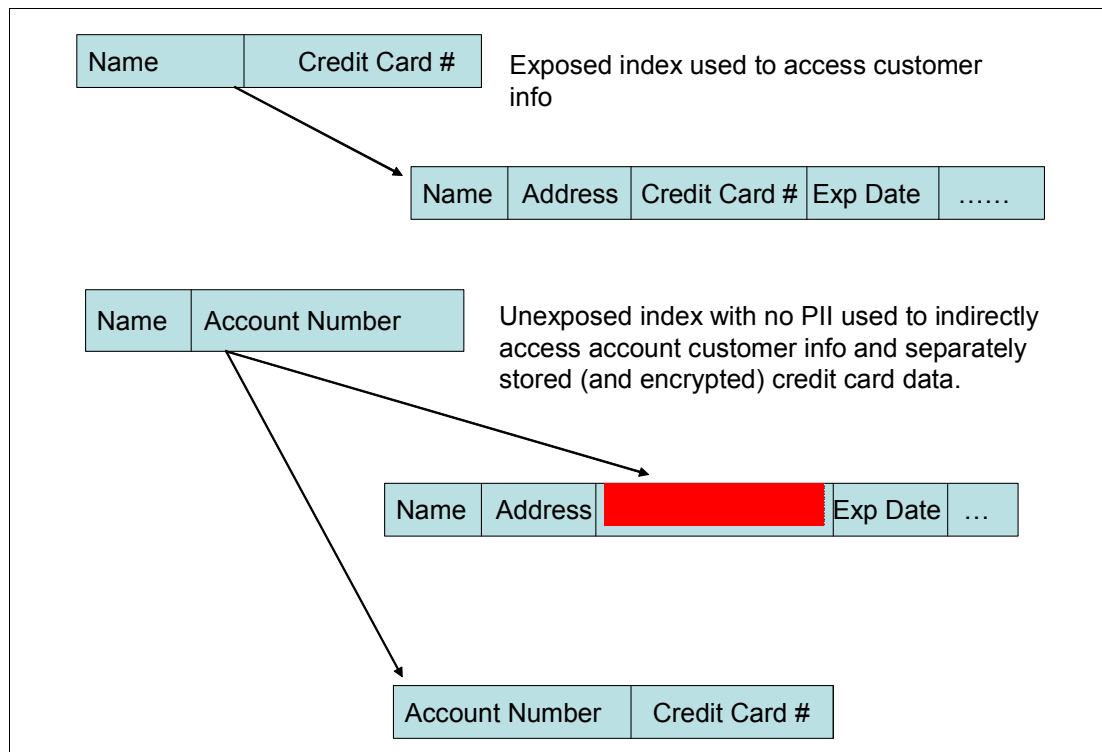


Figure 2-4 Indirect index access representation

As mentioned above, this index includes both PAN and PII columns. In designing an alternative approach, one could create a second table, which would only store the PAN, in this case the credit card number and an account number that could be viewed as a cross-reference number. In the original table schema, the credit card number would be replaced with the account number. Appropriate indexes would be created over each table, and access to the credit card number would occur through the account #. This would

require some application changes, but it would have the desired affect of removing the exposure through the unsecure index. In addition, to conform to the requirement, one could only encrypt the table where the credit card number is stored.

The concept of avoiding an exposed index in database schema design has been around for a number of years, and has been accepted as standard best practice. As part of an encryption implementation, index design needs to be reviewed and indexes which include sensitive data should be identified. With the security and compliance group, these indexes should be classified as exposed or unexposed. For the truly exposed index, consideration needs to be given to the viability of application re-design as described above to provide a remedy that would allow for base table encryption

2.4.4 Encryption standards

As implemented on DB2 for z/OS, there are two different encryption algorithms that can be exploited with the hardware encryption support on System z. These algorithms are the Triple Data Encryption Standard (TDES) and the American Encryption Standard (AES). Both are encryption algorithms that are currently considered secure by industry recognized standards.

To discuss TDES, one must first review its predecessor, DES (also referred to as single DES). Starting in 1974, IBM worked with NIST (National Institute of Standards and Technology) and eventually defined DES (Data Encryption Standard), which became an official encryption standard. DES encrypts and decrypts data in 64-bit blocks, using a 64-bit key (although the effective key strength is only 56 bits). It takes a 64-bit block of plaintext as input and outputs a 64-bit block of ciphertext. The main algorithm is applied to the plaintext 16 times to produce the ciphertext. With the backing of the NIST, DES became the defacto encryption standard and became the prevalent commercially adopted encryption algorithm.

As time progressed, various shortcuts were devised that allowed a DES key to be broken by various brute force attack methods. At the same time, the speed of computing devices increased to the point that the use of a 56 bit key was generally recognized as insufficient to provide adequate protection. By 1997, this situation prompted the NIST to declare that encryption techniques which relied on a DES algorithm were to be considered weak and insecure. Other than in limited situations where systems require only limited security, DES is no longer an acceptable encryption implementation

Because DES was widely used at the time NIST deprecated its use, an expedient solution was to introduce TDES as an alternative. TDES, which is recognized by the NIST as providing secure enough protection for most purposes, is a construction of applying DES three times in sequence. TDES, with the use of three different keys (56-bit each), has an effective key length of 168 bits.

In May 2002, NIST accepted a second encryption algorithm, the Advanced Encryption Standard, known as AES. AES operates on data in 128-bit blocks, and can apply 128-bit, 192-bit, or 256-bit keys in the algorithm. Dependent on the size of the key, the input plain-text data is processed 10, 12, or 16 times to generate the ciphertext result.

Through the year 2030, both TDES and the Advanced Encryption Standard (AES) will coexist as NIST-approved algorithms, allowing for a gradual transition to AES.

To support these emerging encryption technologies, IBM has developed cryptographic coprocessor support on System z that implements a selection of current standards, including TDES (168-bit key) and AES (128, 192, and 256-bit keys).

There are some considerations regarding the use of the different levels of AES that depend on the type of System z processor in use:

- ▶ TDES (168-bit key) well supported in both z9 and z10.
- ▶ AES (128-bit key) hardware support in both z9 and z10
- ▶ AES (168-bit key and 256-bit key) software support on z9
- ▶ AES (168-bit key and 256-bit key) hardware support on z10 only

Note: For the purpose of the encryption features of System z processors, we only focus on the family of z9 and z10 machines. Earlier machine generations have different levels of cryptographic support that exhibit different implementation and performance characteristics.

In general, as implemented in System z processors, hardware encryption exhibits much better performance characteristics than software implementations. Software-supported AES encryption is not recommended.

Subject to the availability of support for the different forms of AES and TDES encryption on the hosting processor, the Data Encryption for IMS and DB2 Databases Tool supports the generation of exits that can exploit both AES and TDES keys.

2.4.5 Cost of security versus SLA

A service level agreement (SLA) in most IT organizations is an agreement where a level of service is defined in concrete form. The SLA creates an understanding about the delivery of services between IT and internal business customers. These services can be defined as measurements for levels of availability, performance, serviceability of existing and new applications, and other attributes. Many times, these agreements require that the IT organizations work with their business customers to verify that the agreed upon measurements are related to services that are viewed as being most important to keeping the business operating in an efficient and profitable manner. Once clearly defined, an SLA can help improve relationships with business customers, can provide both parties use in departmental budget allocations, and can provide improved ability to compete with outsourcing challenges. Properly implemented, SLAs provide metrics for IT performance and can help build more informed internal customers. The benefit of SLA-based measurements can provide the internal business consumer of IT services a direct relationship between levels of support and dollars.

The traditional view of turning on audit collection or implementing encryption in a high volume OLTP database environment is that the collection of audit information can generate significant and in some cases unacceptable overhead. This perception, coupled with a focus on maintaining consistent measurements for response times and availability, as cited in SLA measurement objectives, tends to be the main objection voiced by DBAs when confronted with the prospect of operating in an audited environment. It is for this reason, when embarking on an effort to implement audit-based protection in the database environment, there needs to be some review of existing SLA agreements.

Many times the reason for implementation of an auditing or encryption framework is a causal effect of demands placed on an organization from outside regulators. In most cases, the direct financial costs to an organization by not complying with these demands represents a significant impact to the financial bottom line. It is only reasonable that when looking at the impact of auditing overhead to performance metrics described in the SLA, the direct and indirect costs of compliance be taken into consideration.

One additional impact to the existing SLA is that there is now an expectation that the standard set of services provided by IT to the business customers includes an expectation of providing data protection through facilities such as encryption and auditing. The level of this service can be determined internally, or may be spelled out in explicit terms by external regulations.

As discussed in 2.4.6, “The cost of a data breach” on page 43, there exists a real potential for significant financial impact and harm to an organization in the wake of a data breach. Given the exposure faced by the business, one should be able to adjust service level objectives to accommodate the impact to response time and availability that an implementation of encryption or auditing brings. The discussion of this accommodation needs to include many internal stakeholders, including IT, line of business customers, security and compliance, and legal representatives. There needs to be some understanding of the requirements defined by the various compliance initiatives, and some interpretation of the level of remediation needed to demonstrate conformance. There needs to be some balance struck between one view that could include a demand to audit every access, and the opposite end of the spectrum, which postulates the overhead introduced as too expensive and the resulting do nothing position.

There is some impact to performance associated with security implementations that introduce auditing or encryption into any DB2 for z/OS environment. What is required is a paradigm shift in the way IT organizations and their internal customers view the commitments to SLA components such as response time and availability. The compliance solutions introduce new challenges to the database professional, bringing impact to the daily care and feeding of the DB2 for z/OS ecosystem, increased response times, additional disk resource consumption, incremental additional operational complexity, and more.

What needs to be understood by all stakeholders in this decision is the long term benefit, and the short term impacts to the SLAs. Negotiations to modify goals and measurements need to reflect the ultimate goals. If they can understand the relationship between cost and the level of support, they can make a better business decision about what support is required and how much it costs.

2.4.6 The cost of a data breach

For many organizations, data constitutes their most important and prized asset. Information can describe an institution’s relationship with its customers, competitive or proprietary processes, trading relationships with partners, and tactical and strategic positioning against their competitors. Because of the close interaction with an organization’s business, and the data that describes that business, it becomes a paramount requirement to protect that data. It is for all these reasons and more that when data is lost or stolen, real and significant damage can be incurred.

There have been a number of recent data breach incidents whose short and long term ramifications have been chronicled. Developing an understanding of these ramifications, and the potential costs associated with them can help a company build a true picture of the risk and financial impact of an inadvertent or deliberate data breach. This can be used to assess the cost benefit that comes with implementation of additional security infrastructure, such as encryption and auditing solutions.

One characteristic of data breach events and subsequent application of state statutes, for the majority of states, has dictated that once a data breach occurs, the impacted individuals be notified whenever their personal data has been affected. This can include personal data that has been stolen, lost, or compromised in some way.

It is this requirement for notification, mitigation of the associated follow-on consequences, and application of remedies that directly address the original source of the data breach that comprise what is termed direct costs:

- ▶ Forensic analysis and internal investigation of the theft

After the initial discovery of the incident is the effort to quantify the extent of the breach, understand the avenue used to execute the breach, and the institution of a short term remedy to close the source of the breach and avoid any additional loss of data. Among the important questions that need quick response is to quantify the extent of the exposure, the number of exposed individuals, the nature of information stolen, and to identify the specific population of exposed individuals.

Note: While the requirements for breach notification can vary from state to state, many organizations may not be required to perform customer notification if the data breach involves encrypted data.

- ▶ Notification campaign, emails, phone calls, letters, and so forth

Once the number and identities of the affected customer base has been discovered, each individual must be contacted to inform them of the specific nature of the data breach. This notification might need to include multiple means of contact, and in the lack of accurate forensic information, the community targeted in the notification campaign might be much larger than that directly impacted by the breach.

- ▶ Call center costs due to increased volume of customer traffic

With the public acknowledgement of any data breach, and as the result of a successful notification campaign, there is a marked increase in the number of customer interactions requiring the intervention of the call center. This activity can tax the existing call center infrastructure and interfere with their ongoing support of normal customer services. Temporary increases in call center capabilities need to be provided to manage this increased activity.

- ▶ Legal costs for defence and investigation

To verify proper adherence to regulations and to help protect the organizations legal standing, external communications and response are subject to legal review.

- ▶ Internal investigations resulting in mitigation

Along with the external response to contact and disclose the breach, inward inspection results in changes in procedures, implementation of new technology, and additional personnel to repair and better protect customer information from future breaches. Instead of a measured and planned implementation of these remedies, there is often a rush to judgement that results in poorly planned and executed implementation. This could negatively impact other IT or line of business processes and result in increased use of IT and computing resources.

- ▶ Triage to salvage customer and investor relations

There is increased scrutiny on the post-breach activities of an impacted organizations by the customer base and the investor community. As the details of the data breach become more known and ultimate costs to an enterprise both factual and speculative become known, increased pressure is placed on the business to mitigate the concerns of the existing customer base.

- ▶ Fees and penalties

Depending on the nature of the data breach, and in the presence of regulatory statutes or industry regulations, there could be significant costs levied in the way of fines and penalties. In some instances, there could also be legal ramifications that could result in an organization's executives being subject to incarceration.

A data breach can have significant financial impact. However, much more damage can occur from the second class of damage, which we categorize as indirect costs. This category includes impact to a company's standing in the marketplace, and while the implications might not be readily apparent, the impact of a data breach, real and irrevocable damage can occur.

- ▶ Lost employee productivity

Short term efforts to conduct forensics and implement triage to protect against future breaches require redirection of personnel. Deliverables and strategic initiatives can be delayed by this diversion of talent.

- ▶ Erosion of customer base due to loss of confidence

A significant number of the affected customer population is at risk to terminate their relationship with any organization that allows personal information to be stolen. This erosion of confidence can expand by word of mouth to their network of acquaintances.

- ▶ Reticence for new customers to establish relationships

Along with a loss of existing customers, any public disclosure of a data breach sends a message of a lack of institutional control and an apparent cavalier attitude of data custodianship to potential customers.

- ▶ Reduced shareholder confidence and value

As the financial ramifications of a data breach, and the necessary monetary investment to address and rectify the situation become known, there is a direct impact on the profit and loss position of the company. This can have ramifications that contribute to a perceived value of a company and the value of its stock.

- ▶ Decreased competitive standing

Given the competitive landscape that most companies operate in, there is some significant damage to the brand name of any organization that suffers a public data breach. This presents some meaningful and compelling collateral that is used against a company by their competition. Brand image damage and erosion of the customer base might represent the most expensive downstream effect from a data breach.

2.4.7 ROI calculation

There can be severe financial repercussions that arise from a data breach. However, there is a corresponding cost associated with implementing protections such as encryption or auditing database activity. While sometimes expensive in relation to the existing deployed technology, it is certainly cheaper to analyze the cost benefit and resultant justification before a breach occurs. To assist customers in the quantification of the cost of a typical data breach, IBM has created the IBM Data Governance ROI calculator.

Using documented industry information, the calculator breaks down the potential impact of a data breach into a number of categories, representing both direct and indirect cost categories, which were discussed in the previous section. The information provided by the customer is a number of customers or records that describe information of interest to a potential attacker. In addition, an estimate is provided that describes the percentage of records that become exposed in the event of an exposure. The result is a breakdown of the

lineitem details of the cost of the data breach. Armed with this information, one can clearly substantiate the cost of security remediation against the potential financial damages that come with a data breach.

IBM Data Governance for IMS and DB2 Databases - Cost Benefit Analysis CBA / ROI			
<p>IBM Data Encryption for IMS and DB2 Databases provides you with a tool for both IMS and DB2 in a single product. The tool enables you to leverage the power of Storage Area Networks (SANs) safely while complying with privacy and security regulations in place or being enacted worldwide. During encryption, IMS or DB2 application data is converted to database data that is unintelligible except to the person authorized by your security administrator. Sensitive data is protected at the row level for DB2 and the segment level for IMS.</p>			
<p>Please input all cells that are "beige" in color</p>			
CUSTOMER INPUT		IBM INPUT	
Total number of customer records	5,000,000	Cost of IBM Data Encryption Tool (OTC for 200 msu)	\$0.00
Percent (%) of customer records exposed in a data breach	3%	Annual S&S For IBM Data Encryption Tool	\$0.00
Total number of customer records exposed in a data breach (calculated)	150,000	Cost of IBM Audit Management Expert Tool (OTC)	\$0.00
		Annual S&S For IBM Audit Management Expert Tool	\$0.00
Direct Costs of a Data Breach per Record*			
	per Record	Total	
Free or discounted services	\$26.00	\$3,900,000.00	
Notification letters, phone calls, e-mails, Web, media	\$14.00	\$2,100,000.00	
Legal defense services and criminal investigations	\$7.00	\$1,050,000.00	
Legal, audit, and accounting fees	\$4.00	\$600,000.00	
Call center expenses	\$3.00	\$450,000.00	
Public and investor relations	\$1.00	\$150,000.00	
Internal investigations	\$1.00	\$150,000.00	
Total Direct Costs of a Data Breach	\$56.00	\$8,400,000.00	
Indirect Costs of a Data Breach per Record*			
	per Record	Total	
Lost employee productivity (Employees diverted from other tasks)	\$25.00	\$3,750,000.00	
Opportunity cost (Customer churn and difficulty in getting new customers)	\$50.00	\$7,500,000.00	
Total Indirect Costs of a Data Breach	\$75.00	\$11,250,000.00	
COST SUMMARY			
	per Record	Total	
Direct Costs of a Data Breach per Record	\$56.00	\$8,400,000.00	
Indirect Costs of a Data Breach per Record	\$75.00	\$11,250,000.00	
Total Costs	\$131.00	\$19,650,000.00	
ROI SUMMARY			
Total Cost Savings with product purchase included		\$19,650,000.00	
* Consider HW requirement for DG solution			
CUSTOMER INPUT SOFT SAVINGS			
		\$0.00	
		\$0.00	
		\$0.00	
Total Savings / Value		\$19,650,000.00	
Annual Savings thereafter		\$19,650,000.00	



Figure 2-5 ROI Calculator



Part 2

IBM data governance portfolio

Data governance is a quality control discipline for assessing, managing, using, improving, monitoring, maintaining, and protecting organizational information.

IBM has formed the Data Governance Council, the primary focus of which is to study common challenges and develop best practices in data governance and provides products and services to satisfy the governance needs.

In this part we introduce the IBM information management products that provide the data governance solution.

The chapters are as follows:

- ▶ Chapter 3, “IBM data servers on z/OS” on page 49
- ▶ Chapter 4, “IBM information management tools” on page 91
- ▶ Chapter 5, “Tivoli products” on page 99
- ▶ Chapter 6, “Optim solutions” on page 109



IBM data servers on z/OS

This chapter discusses the security functions provided by data servers and their applicability in securing the data server.

Because we know that systems are built and used in layers, the security also needs to be implemented in each layer. A data server is no longer protected by a secure perimeter. To do business, the enterprises are open to the Web, have remote partners and employees accessing and sharing data, and provide services of varying degree of confidentiality.

The z/OS operating systems and System z platform are the foundation for security. If the foundation is not secure and strong, then the other layers cannot protect themselves. Security is only as good as the weakest link. Remember that if security is not stringent, then access to the operating system facilities must be restricted.

The network must be the focus for an attack and for defense. Data on a LAN is often vulnerable unless encryption is used. Firewalls can prevent some problems, but do not address many others.

Data servers such as DB2 and IMS provide a variety of protection mechanisms. The lower levels have stronger defenses. If application code uses the security mechanisms, the need for assurance is much less. If the application implements security, much stronger assurance is required. The application needs to pass through the security information in order not to compromise the ability to use data servers and z/OS security.

In this chapter, we have grouped the data servers functions in the four areas defined in the Data Server Blue Print:

- ▶ Authentication
- ▶ Authorization
- ▶ Encryption
- ▶ Auditing

In this chapter we provide an overview of how the functions available to the major z/OS data servers can protect their resources. We look at the following features:

- ▶ DB2
- ▶ IMS
- ▶ VSAM

3.1 Security categorization

Security control components for data servers are often categorized as follows:

- ▶ Identification and authentication determine who the user is.
- ▶ Access control uses that identification to determine what resources can be used.
- ▶ Confidentiality and privacy controls help ensure that the access is allowable and monitored.
- ▶ Data integrity controls are the basis for every database management system, with the ACID properties (atomicity, consistency, isolation, and durability).
- ▶ Non-repudiation assures that authorized users are not denied access.
- ▶ Encryption has to be considered for network passwords and confidential data.
- ▶ Audit is the step of assuring that the access controls are working as intended. This step lets us correct the inevitable mistakes and find attacks.
- ▶ Security management is the process of setting up the controls.

Whenever possible, the authorization and ownership should be to or by a group or a role, but the individual who requests information needs to be understood to provide accountability.

The principle of *least privilege* should be applied throughout. This principle requires that each subject in a system be granted the most restrictive set of privileges (or lowest clearance) needed for the performance of authorized tasks. The application of this principle limits the damage that can result from accident, error, or unauthorized use.

One of the primary concepts in security is giving the individuals only the privileges needed to do the job. That often means using more granular controls.

Another key principle is ease of safe use. We want individuals to have all of the privileges they need to do the full job, but no more. If there is less complexity in the security controls, that means less cost and generally results in better compliance.

3.1.1 Data servers security areas

As described in 2.1, “The IBM Data Server Security Blueprint” on page 20, data server security capabilities have been assigned to four broad areas of authentication, authorization, encryption, and auditing.

▶ Authentication

Authentication is the use of multiple mechanisms for validating the identity of a requestor. The most common implementation is to challenge the requestor for a user ID and password. The user ID and password are static entities that generally do not change for an extended period of time, although some security policies can enforce a relatively frequent password change. The major drawback of such an approach is that if a user ID and password are compromised, other users can easily assume their identity. For non-encrypted user IDs and passwords that flow across a TCP/IP network, it is relatively easy to discover the user ID and password. There are implementations of authentication other than user ID and password challenges. A less secure approach is simply a user ID without the requirement for a password. A more secure approach includes authentication standards such as Kerberos. In Kerberos implementations, a third-part authentication server in conjunction with public and private keys and a timestamp are able to allow a requestor and server to verify each other's identity. The important notion about Kerberos is that the tokens used to authenticate a user are not static. Each time an exchange is

created the actual credentials exchanged are based upon a number of public and private keys and a timestamp. That means that even if the security ticket was intercepted, it would be extremely difficult to derive value out of it.

► Authorization

Authorization is the next security control encountered. It is the process of controlling access to resources based upon the authentication of a user. As part of the system, a security policy dictates what resources a specific identity can access and what type of access is allowed. For example, when a user must be authorized to access a table for the intended action (select, insert, update, delete). If the user is only authorized to select a table, attempts to insert, update, or delete will be denied, but attempts to select will be permitted

► Encryption

Encryption is the process of transforming plain text data information by using an algorithm which makes it unreadable to anyone except those authorized to use the key to decrypt it. The intent is to protect the confidentiality of information. There are different layers in the handling of data and for each one of them encryption should be considered.

For transmitting data, there are two options:

- Native data stream encryption supported in the database protocols and Secure Sockets Layer (SSL) supported in the network layer. For SSL, DB2 for z/OS exploits z/OS Communications Server's Application Transparent Transport Layer Support (AT-TLS).
- The native data stream encryption uses DES to provide a level of performance over SSL.

For data on disk encryption, there are also two options:

- The native encryption and decryption column functions provided by the DB2 for z/OS
- The IBM Data Encryption for IMS and DB2 Databases Tool used to encrypt rows.

When offloading backups and archive logs, the tape units offer encryption built-in to the drive to protect the archive tape. All exploit System z Crypto hardware features designed to provide better performance and industry level security built-in to z/OS.

► Auditing

Auditing is the process of recording actions taken by users. This includes actions that are both permitted and denied. A good auditing implementation will allow review of all actions by users. Not only are the actions recorded, but the service that authenticated the user and allowed or denied the user authorization to access a resource must all be recorded. Auditing logs must also be blocked from update by users of the system. The ability to remove an auditing entry would invalidate the benefit of the auditing system.

3.2 DB2

This section discusses how DB2 for z/OS implements the security categories mentioned 2.1, "The IBM Data Server Security Blueprint" on page 20.

3.2.1 Authentication

Authentication is the first security capability encountered when a user attempts to use the DB2 for z/OS product. The user must be identified and authenticated before being allowed to use any of the DB2 for z/OS services.

The primary job of identification and authentication in DB2 is assigned to the security subsystem. DB2 for z/OS uses the z/OS Security Server (RACF or equivalent) for authentication and authorization to access any DB2 subsystem. This technique means that access for many resources can be more consistent, whether the resource is a file, a printer, or communications or database access.

Another way of distinguishing the level of security is the software layer used for the access control. The tightest security would use a single security monitor. Using system controls and subsystems will allow tighter security than having application programs provide the security.

There are many different environments for DB2, with different connections and security. DB2 uses the security context when possible, so batch jobs, TSO users, IMS, and CICS transactions have security that uses a consistent identification and authentication. This is true for stored procedures from these environments as well. The large number of options, exits, environments, and asynchronous or parallel work provide challenges for security. Some key applications manage security at the application level.

For some work, such as distributed database serving, DB2 is the initial point on this platform. For this work (DRDA® distributed access, remote stored procedures), DB2 establishes the security context and manages the work

RACF-managed security

RACF is used for the following purposes:

- ▶ Control connections to the DB2 subsystem.

The ability of a user or address space to connect to DB2 is controlled through checks in the DSNR RACF resource class.

- ▶ Assign identities:

- The DB2 primary authorization ID is a RACF identity.
- Secondary authorization IDs are often derived by a sign-on exit routine from the RACF-generated list of groups.

- ▶ Protect the underlying DB2 data store.

The underlying data sets of DB2 can be protected by RACF.

In addition to DB2-provided security, use RACF to control access to DB2 objects, authorities, commands, and utilities by making use of the RACF access control module. This RACF access control module is activated at the DB2 access control authorization exit point DSNX@XAC by replacing the default DB2-provided exit routine by a version that allows RACF checking.

The RACF access control module provides a mechanism to perform the following tasks:

- Control and audit resources for multiple DB2 subsystems from a single point.
- Define security rules before a DB2 object is created.
- Preserve security rules for dropped DB2 objects.
- Protect multiple DB2 objects with a single security rule using a combination of RACF generic, grouping, and member profiles.
- Validate a user ID before giving it access to a DB2 object.
- Preserve DB2 privileges and administrative authorities.
- Provide flexibility for multiple DB2 subsystems with a single set of RACF profiles.
- Administer DB2 security with a minimum of DB2 skills.
- Eliminate the DB2 cascading revoke.

The RACF access control module is invoked in the following circumstances:

- Once at DB2 subsystem startup, to perform any required set up prior to authorization checking. Authorization profiles are loaded during this invocation.
- For each DB2 authorization request. This corresponds to the point when DB2 accesses security tables in the catalog to check authorization on privileges.
- Once at DB2 subsystem termination to perform cleanup before DB2 stops.

DB2 provides a default exit, DSNX@XAC. This version of the exit routine returns a code to its invoker, indicating that a user-defined access control authorization exit is not available.

Note: Starting in DB2 V8, the RACF access control module DSNX@XAC is provided by DB2. In earlier versions, the exit was provided by RACF.

DB2 provides the following two sources in SDSNSAMP:

- DSNXSXAC
Dummy default version. A compiled version is present in SDSNLOAD for use when not using external security.
- DSNRXAC
Sample version that allows the use of RACF for external control of DB2 resources. A compilation is required to replace the default version.

Attach: IMS, CICS, and RRS authentication

For requests from IMS-dependent regions, CICS transaction subtasks, or RRS connections, the initial primary ID is not obtained until just before allocating a plan for a transaction. A new sign-on request can run the same plan without deallocating the plan and reallocating it. Nevertheless, the new sign-on request can change the primary ID. Unlike connection processing, sign-on processing does not check the RACF user ID of the address space.

DB2 determines the initial primary ID as follows:

- ▶ For CICS transactions, the ID used as the primary authorization ID is determined by attributes in the DB2CONN or DB2ENTRY definitions, depending on the thread type.
- ▶ For IMS sign-on from message-driven regions, if the user has signed on, the initial primary authorization ID is the user's sign-on ID. IMS passes to DB2 the IMS sign-on ID and the associated RACF connected group name, if one exists. If the user has not signed on, the primary ID is the LTERM name, or if that is not available, the PSB name.
- ▶ For a batch-oriented region, the primary ID is the value of the USER parameter on the job statement, if that is available. If that is not available, the primary ID is the program's PSB name.

DB2 then runs the sign-on exit routine. Using the IBM-supplied default sign-on exit routine ensure that the initial primary authorization ID remains the primary ID, the SQL ID is set equal to the primary ID, and no secondary IDs exist.

As for connection processing, if you want the primary authorization ID to be associated with DB2 secondary authorization IDs, you must replace the default sign-on exit routine.

The sample sign-on routine sets the initial primary authorization ID unchanged as the DB2 primary ID, and the SQL ID is made equal to the DB2 primary ID. If RACF is not active, no secondary IDs exist. If RACF is active but its list of group's option is not active, one secondary ID exists. This is the name passed by CICS or by IMS. If RACF is active and you selected the

option for a list of groups, the routine sets the list of DB2 secondary IDs to the list of group names to which the RACF user ID is connected. The list of group names includes the default connected group name.

DB2 JDBC authentication

For JDBC™ applications, there are multiple choices for authenticating a user. These options are dependent upon the connection mechanism, either type 2 or type 4. The options allow DB2 to either inherit the previously authenticated RACF user ID of the process connecting to DB2, allow the connecting thread to specify only a user ID, or specify a user ID and password.

DB2 Universal type 4 Driver authentication

For remote requests, a combination of settings at the type 4 JDBC Driver and the DB2 for z/OS server's Distributed Data Facility determine the types of authentication that are supported.

You choose which type of authentication to use when attempting to connect from the JDBC client through a property of the datasource. The property, `securityMechanism`, determines security attributes for transmitting data in addition to determining the requirements for authentication. Here are the supported types for the JDBC driver's `securityMechanism` property, which is of type integer:

▶ **CLEAR_TEXT_PASSWORD_SECURITY**

In this case the user ID and password are required to be sent to the server for authentication, and the fields are passed through the network in unencrypted text.

▶ **USERID_ONLY_SECURITY**

Only the user ID is required to be sent to the server. This value requires that the server's TCPALVER ZPARM must be set to YES (a setting generally discouraged because this results in no password challenge).

▶ **ENCRYPTED_PASSWORD_SECURITY**

User ID and password are required to be sent to the server for authentication. The password is sent encrypted and is not in plain text across the network.

▶ **ENCRYPTED_USER_AND_PASSWORD_SECURITY**

User IDs and passwords are required and are encrypted when sent in the network.

▶ **ENCRYPTED_USER_AND_DATA_SECURITY**

The user ID and password are required for authentication, the user ID and data are sent encrypted, and the password is sent in unencrypted text.

▶ **ENCRYPTED_USER_PASSWORD_AND_DATA**

User ID and password are required and encrypted; data is encrypted as well.

▶ **KERBEROS_SECURITY**

Kerberos infrastructure is exploited; no user IDs or passwords are sent.

DB2 Universal type 2 Driver authentication

For type 2 connections, a connection can be created using a specific authentication by specifying a user ID and password. If a user ID and password are specified, DB2 verifies the values by issuing a SAF call to RACF. If successful, the thread carries the security context of the authorization ID and DB2 associates the thread with the AUTHID while performing actions in DB2. Alternatively, the thread can connect to DB2 without specifying a user ID and password. When this occurs, the thread will assume the security contexts of the process, which is connecting to DB2.

Network-trusted context

Trusted context addresses the problem of establishing a trusted relationship between DB2 and an external entity, such as a middleware server, such as the following servers:

- ▶ IBM WebSphere Application Server
- ▶ IBM Lotus® Domino®
- ▶ SAP® NetWeaver
- ▶ Oracle® PeopleSoft®
- ▶ Siebel® Optimizer

Before trusted context, for connections from these application servers, one system user ID is used to establish the connection and that system user ID performs transactions on behalf of all users. Hence, in DML with unqualified SQL, the default schema was set to the current SQL ID value, which is the primary authorization ID of the system user ID.

You have the option to set a system parameter which indicates to DB2 that all connections are to be trusted. It is unlikely that all connection types, such as DRDA, RRS, TSO, and batch from all sources will fit into this category. It is likely that only a subset of connection requests for any type and source may be trusted or that you want to restrict trusted connections to a specific server. More granular flexibility will allow for the definition of trusted connection objects.

DB2 9 for z/OS provides new options for tighter security and allows for more granularity and additional flexibility. You now are able to create two new database entities: trusted context and role.

A trusted context establishes a trusted relationship between DB2 and an external entity, such as a middleware server or another DB2 subsystem. At connect time, a series of trust attributes are evaluated to determine if a specific context can be trusted. After a trusted connection is established, one of the new abilities is to acquire, through a role, a special set of privileges for a DB2 authorization ID within the specific connection that are not available to it outside the trusted connection.

For transactions coming from middleware servers (others than IMS and CICS) or utilities, during the sign-on processing, DB2 checks to see whether the request is from a trusted connection performing a switch user, and if so, whether the primary authorization ID is permitted to switch. If the primary authorization ID is not allowed to switch, the connection is terminated. With trusted context switch user capability, the primary authorization ID changes for every user transaction and hence also the default schema value for unqualified SQL. Many applications or servers may have to change to set CURRENT SCHEMA to the role name for every user transaction. Also, it is a lot of overhead to send SET CURRENT SCHEMA every time to change the qualifier to a role.

The AND QUALIFIER is added to the ROLE AS OBJECT OWNER clause in the trusted context definition. This indicates that rolename will be used as the default for CURRENT SCHEMA special register and will be included in the SQL PATH (in place of USER).

Currently, trusted context attribute JOBNAME does not accept wildcard characters. This creates a problem when attempting to define the jobname for a process that is invoked under a telnet session on z/OS. UNIX® System Services may span a new process to invoke the job and generate a new jobname by appending a number to the logged in user ID. When using the JDBC T2 connection the trusted context expects to have a matching jobname so every possible jobname that UNIX System Services may generate would have to be listed as an attribute.

The attribute JOBNAME is modified to allow for specifying a wild card character as the last character (*) of a jobname-value. Currently, users allowed to switch in a trusted connection are stored in SYSIBM.SYSCONTEXTAUTHIDS table and DB2 validates if the user is allowed to switch. With this approach, users are not visible to RACF administrators and if a user leaves the company it becomes difficult to manage the resources assigned to the user. A new keyword, EXTERNAL SECURITY PROFILE profile-name is added to the trusted context user clause definition to permit the users allowed to switch in a trusted connection to the profile-name defined in RACF.

Once defined (see Example 3-1), connections from specific users through defined attachments and source servers will allow trusted connections to DB2. The users defined in this context can also be defined to obtain a database role.

Example 3-1 At the DB2 server

```
CREATE CONTEXT WAS1
SYSTEM USERID WASPROD
ADDRESS MY.WAS.SERVER
ALLOW USER
JOE WITHOUT AUTHENTICATION,
SAM WITHOUT AUTHENTICATION;
```

A trusted connection can be established for a local or a remote application. The attributes used to establish a trusted context are different for remote versus local applications. For a remote trusted connection, the SYSTEM AUTHID is derived from the system user ID provided by an external entity (for example, a middleware server), when initiating the connection.

- ▶ For a remote trusted connection, the attributes considered are as follows:
 - ADDRESS
IP address or domain name. The protocol is restricted to TCP/IP only.
 - SERVAUTH
A resource in the RACF SERVAUTH class. TCP/IP network resources (IP addresses) can be mapped to RACF SERVAUTH security zones. Using the RACF SERVAUTH class, it is possible to protect access to those network resources.
 - ENCRYPTION
Minimum level of encryption for the connection:
 - NONE: No encryption. This is the default.
 - LOW: DRDA data stream encryption.
 - HIGH: Secure Sockets Layer (SSL) encryption.
- ▶ For a local trusted connection, the SYSTEM AUTHID is derived as follows:
 - Started task (RRSAF): JOB statement USER or RACF USER
 - TSO: TSO logon ID
 - BATCH: JOB statement USER
- ▶ For a local trusted connection, the attribute considered is JOBNAME, which is derived as follows:
Started task (RRSAF): JOB or started class name
 - TSO: TSO logon ID
 - BATCH: JOB name

Trusted contexts cannot be defined for CICS and IMS. DB2 online utilities can run within a trusted connection.

DB2 9 implements new DDL

- ▶ CREATE TRUSTED CONTEXT
- ▶ ALTER TRUSTED CONTEXT
- ▶ DROP TRUSTED CONTEXT
- ▶ COMMENT ON TRUSTED CONTEXT
- ▶ GET DIAGNOSTICS

A new value for DB2_AUTHENTICATION_TYPE. T indicates trusted context authentication.

DB2 9 uses new catalog tables

- ▶ SYSIBM.SYSCONTEXT contains the context name, context ID, and other context related information.
- ▶ SYSIBM.SYSCTXTTRUSTATTRS holds the attributes for a given context.
- ▶ SYSIBM.SYSCONTEXTAUTHIDS stores the authorization IDs that can be *switched to* in a trusted connection.

How a trusted connection comes alive and ends

A trusted connection is a database connection that is established when the incoming connection attributes match the attributes of a unique, enabled trusted context defined at the server. The trust attributes identify a set of characteristics about the specific connection that are required for the connection to be considered a trusted connection.

The relationship between a connection and a trusted context is established when the connection to the server is first created. That relationship remains for the life of that connection. If a trusted context definition is altered, the changed attributes of the trusted context take effect when the next new connection request comes in, or a switch user request is issued within an existing trusted connection.

3.2.2 Authorization

The DB2 configuration has a few parameters with required settings for security, but many other parameters are important to consider. The options, exits, and interfaces can make a big difference. Data set protection is provided by the security subsystem. Software currency can also be important for security.

When an application gains access to a subsystem, the user has been authenticated and access to DB2 for z/OS is checked using RACF. DB2 for z/OS controls access to data through the use of identifiers associated with the authenticated user. A set of one or more DB2 for z/OS identifiers, called authorization IDs, represent the user on every process that connects to or signs on to DB2 for z/OS. These IDs make up the SQL ID. The SQL ID and role, if running in a trusted context, are used for authorization checking within the DB2 database system.

Access to DB2 for z/OS requires the use of packages. Packages are required to execute SQL statements¹. Packages have an owner ID or role associated with them. The owner might be different from the SQL ID or role executing the package. To execute any SQL statements bound in a package, the SQL ID or role associated with the package must have the execute privilege on the package. The package owner is used for privilege checking for any static SQL statements in the package. When executing a dynamic SQL statement, the SQL ID or role must be authorized to perform the action against the DB2 database system, not the owner.

¹ Plans with DBRMs are still usable but are being deprecated, see *DB2 9 for z/OS: Packages Revisited*, SG24-7688.

This allows DB2 for z/OS to perform as much authorization checking when the package is created and not every time it is used. This approach eliminates the need to authorize all users to all objects used in a package.

You can take advantage of mandatory access control in the DB2 database system to protect table data based on the security labels of the rows. When a user accesses a row or a field in the row with an SQL statement, DB2 for z/OS calls RACF to verify that the user is allowed to perform the type of access that is required for the SQL statement. The access is allowed only if the user has the requested access right to all of the rows containing fields that are accessed as part of the SQL statement. For all fields that the SQL statement accesses, DB2 for z/OS checks the security label of the row containing the field and denies access when the user's security label does not dominate the security label of the any one of the rows containing the fields.

A powerful security enhancement in DB2 9 for z/OS is the introduction of trusted contexts, a feature that supplies the ability to establish a connection as trusted when connecting from a certain location or job. Having established a trusted connection, it provides the possibility of switching to other user IDs, thus giving the opportunity of taking on the identity of the user associated with the SQL statement. In addition, it is possible to assign a role to a user of a trusted context. The role can be granted privileges and can represent a role within the organization in that it can hold the sum of privileges needed to perform a certain job, application, or role.

These two constructs together supply security enhancements for a variety of different scenarios ranging from any three-tier layered application, such as SAP, to the daily duties of a DBA maintaining the DB2 for z/OS subsystem.

Authorization IDs for accessing data within DB2

Data access in a local DB2 system can be from a batch program, from a user on an interactive terminal session, or from a CICS or IMS transaction. For the purposes of security, DB2 uses the term *process* to represent all forms of access to data. See Figure 3-1.

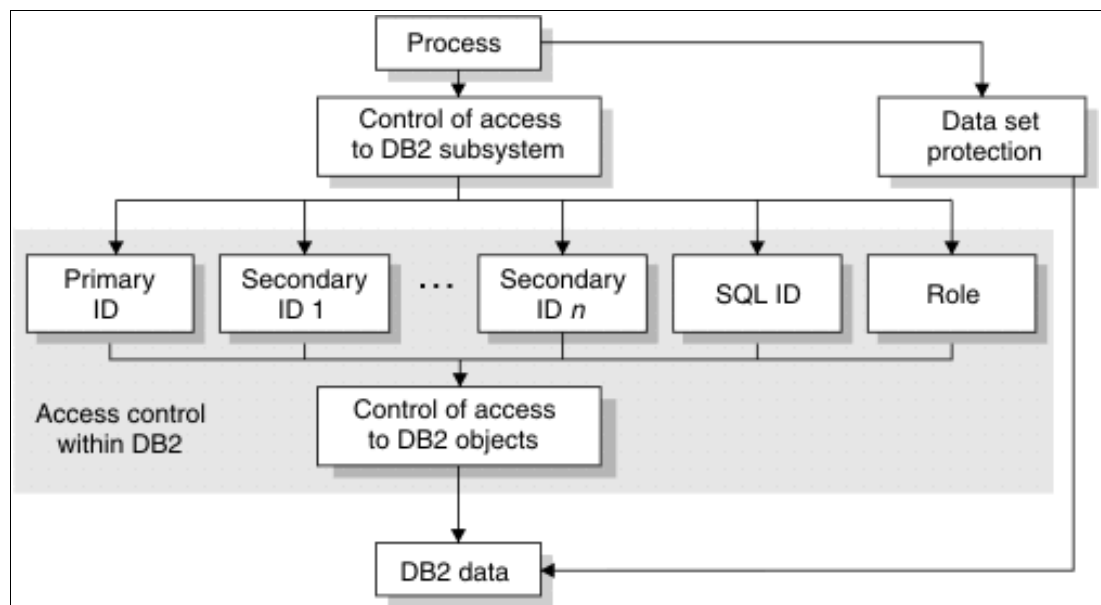


Figure 3-1 Associating IDs to process

Every process that connects to or signs on to DB2 is represented by a set of one or more authorization IDs. When authorization IDs are assigned, every process receives at least one primary authorization ID and one or more secondary IDs, and one of those IDs is designated as the current SQL ID. Also, from within trusted context, a role might be assigned.

The following terms are relevant to the discussion of processes:

- ▶ Primary authorization ID

Generally, the primary authorization ID identifies a process. For example, statistics and performance trace records use a primary authorization ID to identify a process.

- ▶ Secondary authorization ID

A secondary authorization ID is optional. It can hold additional privileges that are available to the process. For example, a secondary authorization ID can be a RACF group ID.

- ▶ SQL ID

An SQL ID holds the privileges that are exercised when certain dynamic SQL statements are issued. The SQL ID can be set equal to the primary ID or any secondary ID.

- ▶ Role

A role is available within a trusted context. You can define a role and assign it to authorization IDs in a trusted context. When associated with a role and using the trusted connection, an authorization ID inherits all the privileges granted to that role. The role, together with the trusted context, is a new database object introduced in DB2 9 for z/OS.

During connection processing and sign-on processing, DB2 sets the primary and secondary authorization IDs for the process to use in the DB2 address space. By default, DB2 uses the authorization IDs that the process has provided. Next to default procedures, authorization IDs can also be assigned to those processes by user-written exit routines.

DB2 has two exit points for authorization routines, one in connection processing and one in sign-on processing. Both exit points perform crucial steps in the assignment of values to primary IDs, secondary IDs, and SQL IDs. DB2 has a default connection exit routine and a default sign-on exit routine. You can replace these with your own exit routines.

During connection processing, DB2 calls RACF to check whether the ID is authorized to use one of the following resources:

- ▶ DB2 resource class DSNR
- ▶ DB2 subsystem

If RACF is active and has verified the RACF user ID, DB2 runs the connection exit routine. Running with the default DB2-provided exit routine makes sure that the default user ID is the primary authorization ID, no secondary IDs exist, and the SQL ID is the same as the primary ID.

If you want to use DB2 secondary authorization IDs, you must replace the default connection exit routine. The connection authorization exit routine must be named.

DSN3@ATH. DB2 installation job DSNTIJEX in SDSNSAMP provides a step to replace the default connection exit with a sample connection exit routine, of which you can find the source in the SDSNSAMP library in member DSN3SATH. The sample connection exit routine sets the DB2 primary ID and the SQL ID in the same way as the default routine.

The setting of the secondary authorization ID depends on RACF options. There can be no secondary authorization ID at all, one single ID (the default connected group name), or a list of secondary authorization IDs equal to the list of group names to which the RACF user ID is connected.

If the default connection exit routine and the sample connection exit routine do not provide the flexibility and features that your subsystem requires, you can write your own exit routine.

As a final step of connection processing, DB2 determines whether the connection is to be established as trusted. If a trusted context with a system authorization ID matching the primary authorization ID exists, and the attributes of the trusted context match those of the connection request, the connection is established as trusted.

DB2-provided security

DB2 resources access control can be managed through a DB2 built-in mechanism.

You can use the SQL GRANT or REVOKE statements to grant and remove privileges if you enable authorization checking during the DB2 installation. You can grant or revoke privileges to and from authorization IDs or roles if running in a trusted context. You can only revoke privileges that are explicitly granted.

You can grant privileges in the following ways:

- ▶ Grant a specific privilege on one object in a single statement.
- ▶ Grant a list of privileges.
- ▶ Grant privileges on a list of objects.
- ▶ Grant ALL, for all the privileges of accessing a single table, or for all privileges that are associated with a specific package.

For example, grants control the use of buffer pools, utilities, storage groups, DB2 commands, and so on. All grants are recorded in the DB2 catalog tables as follows:

- ▶ SYSCOLAUTH (Privileges that are held by users on column level)
- ▶ SYSTABAUTH (Privileges held by users on the table level)
- ▶ SYSDBAUTH (Privileges held by users over databases)
- ▶ SYSUSERAUTH (System Privileges held by users)
- ▶ SYSRESAUTH (Privileges on resources buffer pool, Storage group)

The GRANT statement grants privileges to authorization IDs or roles. Privileges can be explicit and implicit.

Explicit privileges have names and are held as the result of GRANT and REVOKE statements. There is a set of specific privileges for each type of DB2 object.

Examples of GRANT are as follows:

- ▶ GRANT SELECT ON TABLE SYSIBM.SYSDATABASE TO USER1
- ▶ GRANT SELECT ON TABLE SYSIBM.SYSUSERAUTH TO PUBLIC

The first statement gives USER1 the authority to select from table SYSIBM.SYSDATABASE. The second statement allows all users to select rows from SYSIBM.SYSUSERAUTH. USER1 can be a TSO user ID or a RACF group representing a list of users.

Sets of privileges are grouped into administrative authorities. These authorities form a hierarchy. Each hierarchy includes a specific group of privileges.

The administrative authorities, shown in Figure 3-2 on page 61, fall into the categories of system, database, and collection authorities.

In the middle of Figure 3-2 on page 61, we have the highest ranking administrative authority, SYSADM, which includes all DB2 privileges except a few that are reserved for installation SYSADM

On the right side of Figure 3-2 we have database administrator authority, which includes the privileges required to control a database. Users with DBADM authority can access tables and alter or drop table spaces, tables, or indexes in that database.

On the left side of Figure 3-2, we have SYSCTRL, which includes the privileges to issue DB2 commands and to terminate utilities.

Each high level of authority includes the privileges of all lower-ranking authorities.

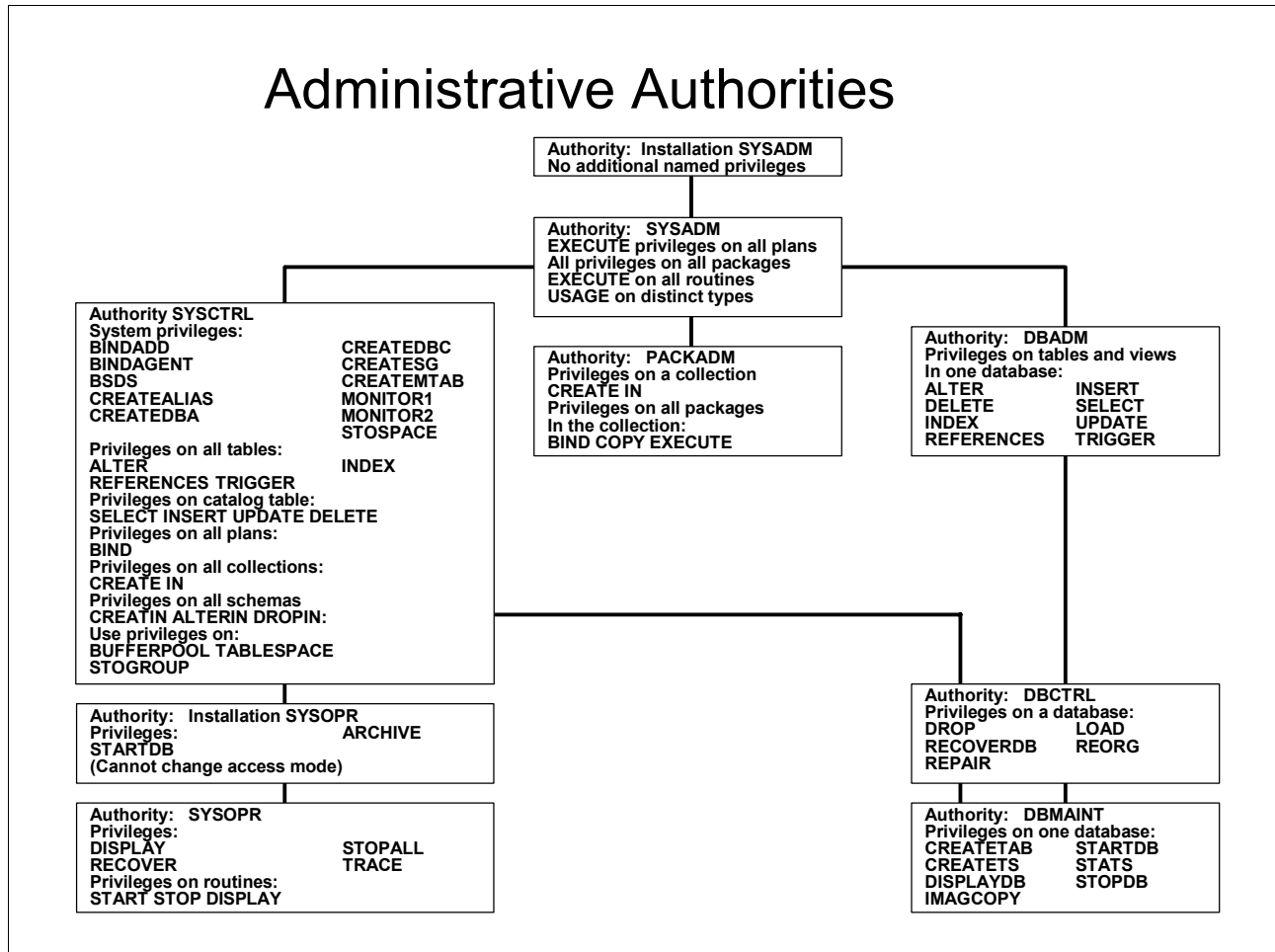


Figure 3-2 DB2 privileges

Examples of the usage of administrative authorities are:

- ▶ GRANT SYSOPR TO USER1
- ▶ GRANT DBADM ON DATABASE RACFDB2 TO USER1

There are two additional DB2 administrative authorities. One or two IDs are assigned the installation SYSADM authority, and one or two IDs are assigned the installation SYSOPR authority. These are similar to SYSADM and SYSOPR, respectively. DB2 does not record these authorities in the catalog, but they are defined in the subsystem initialization parameter module DSNZPARM.

No other ID can revoke these installation authorities. These authorities are also allowed to perform some special actions such as running the CATMAINT utility, accessing DB2 when the subsystem is started with ACCESS(MAINT), or starting the directory and catalog databases

when they are stopped or in restricted status and running all allowable utilities against these databases. For further details about the installation SYSADM and installation SYSOPR administrative authorities, see *DB2 Version 9.1 for z/OS Administration Guide*, SC18-9840.

Implicit privileges are related to the ownership of an object. Ownership is set at object creation time. When the user is the owner of the DB2 object, the user implicitly holds certain privileges over that object. The implicit privileges of ownership are different for each different object.

As an example, for a table, these are the privileges:

- ▶ Alter or drop the table or any indexes on it
- ▶ Lock the table, comment on it, or label it
- ▶ Create an index or view for the table
- ▶ Select or update any row or column
- ▶ Insert or delete any row
- ▶ Use the LOAD utility for the table
- ▶ Define referential constraints on any table or set of columns
- ▶ Create a trigger on the table

There are a few requirements for basic DB2 security. One is that authorization should always be used. The number of people who can be install SYSADM, SYSADM, or SYSCTRL should be small. These names should be groups or roles, and the number of people able to use those groups or roles should be small. A rule of thumb would be 10 people, most of whom do not use this authority for their normal work. Other administrative users should be restricted to the access needed and also be groups or roles. Where the work is sensitive, auditing is required.

BINDAGENT and SYSCTRL are relatively weak security. The BINDAGENT privilege is intended for separation of function, not for strict security. A bind agent with the EXECUTE privilege might be able to gain all the authority of the grantor of BINDAGENT. One of the keys to success is keeping the security as simple as possible. Having a direct mapping from the security policy to the implementation will keep mistakes to a minimum, but we must allow for mistakes and for correcting those mistakes.

Access with SYSADM or SYSCTRL authority should be audited.

SYSOPER can be controlled.

MONITOR2 should only be provided to those who can view all work on the DBMS. Public access should be avoided without careful justification and understanding of the security policy.

There is a fairly new example of how to provide EXPLAIN access when the individuals do not have direct access to the data. An example showing how to have EXPLAIN authorization without direct access to the tables is provided with the V8 Visual Explain² and APARs PQ90022 and PQ93821. For this example, you may not want the binder to have SYSADM, and will not want to grant access to public.

<http://www.ibm.com/software/data/db2/zos/osc/ve/index.html>

For a user to have implicit authority over an object, the object owner needs to be either that user's primary authorization ID, one of the user's secondary authorization IDs, or the name of a role associated with the user in a trusted context.

² DB2 Visual Explain is deprecated. For DB2 for z/OS Version 8 and DB2 9 for z/OS, DB2 Visual Explain is replaced by the following offerings: IBM Optimization Service Center for DB2 for z/OS, IBM DB2 Optimization Expert for z/OS, IBM Data Studio

Note: When an object is created within a trusted context by an authorization ID with a role associated, and this role is defined with `ROLE AS OBJECT OWNER`, the role is the owner of the object regardless if the object is qualified or not.

Roles

Roles are used to provide a more flexible technique than groups or users in assigning and controlling authorization, while improving consistency with the industry. A network trusted context provides a technique to work with other environments more easily, improving flexibility.

A database role is a virtual authorization ID that is assigned to an authid through an established trusted connection. Within a trusted connection, DB2 allows only one role to be associated with a thread at any point in time. A role is a database entity to which one or more DB2 privileges can be granted to or revoked from. Roles provide a means to acquire context-specific privileges. To support roles in a trusted context, DB2 extends the `GRANT` and `REVOKE` statements to add roles to the list of authorization names to which privileges are granted and revoked. Privileges can be granted and revoked by a role within an established trusted connection using the `GRANT/REVOKE` statements are modified as follows:

- ▶ `GRANT/REVOKE` (collection privileges) `TO ROLE` role-name
- ▶ `GRANT/REVOKE` (database privileges) `TO ROLE` role-name
- ▶ `GRANT/REVOKE` (distinct type or JAR privileges) `TO ROLE` role-name
- ▶ `GRANT/REVOKE` (function or procedure privileges) `TO ROLE` role-name
- ▶ `GRANT/REVOKE` (package privileges) `TO ROLE` role-name
- ▶ `GRANT/REVOKE` (plan privileges) `TO ROLE` role-name
- ▶ `GRANT/REVOKE` (schema privileges) `TO ROLE` role-name
- ▶ `GRANT/REVOKE` (sequence privileges) `TO ROLE` role-name
- ▶ `GRANT/REVOKE` (system privileges) `TO ROLE` role-name
- ▶ `GRANT/REVOKE` (table or view privileges) `TO ROLE` role-name
- ▶ `GRANT/REVOKE` (use privileges) `TO ROLE` role-name

Roles can create and own objects. If specified in the trusted context definition (`ROLE AS OBJECT OWNER`), a role becomes the owner of objects created in a trusted connection. Roles must have all the privileges necessary to create the objects. If a role owns a created object, the user requires a `GRANT` to access it outside the context.

The role can be assigned and removed from individuals through the trusted authorization context as needed. This allows a DBA to perform object maintenance during a change control window on a Saturday night, for example. But when Monday arrives, they do not have the authority to do this same work. See Example 3-2.

Example 3-2 Defining a role

```
CREATE ROLE PROD_DBA;  
GRANT DBADM ... TO PROD_DBA;  
CREATE TRUSTED CONTEXT DBA1 ...  
DEFAULT ROLE PROD_DBA WITH ROLE AS OBJECT OWNER AND QUALIFIER;
```

Auditing trails of the work completed during the maintenance window are available for verification by a security administrator or auditor.

DB2 extends the trusted context concept optionally to assign a default role to a trusted context and optionally to assign a role to a user of the context.

Catalog tables

There are two new catalog tables to support roles:

- ▶ SYSIBM.SYSROLES contains one row for each role.
- ▶ SYSIBM.SYSOBJROLEDEP lists the dependent objects for each role.

Cascading revoke

When access is revoked from a certain DB2 authorization ID, all access granted by that authorization ID to the same resource is revoked as well. Assume the following scenario:

- ▶ User U1 grants a privilege P to user U2 with GRANT option on an object.
- ▶ User U2 grants the same privilege P to user U3 on the same object.

When U1 revokes the privilege P from user U2, user U3 also loses its privilege on that object.

Role

You can revoke all privileges that are assigned to a role by simply dropping the role itself or using the REVOKE statement. When you attempt to drop a role, make sure that the role does not own any objects. If the role owns objects, the DROP statement is rolled back. If the role does not own any objects, the role is dropped. As a result, all privileges held by this role are revoked and the revocation is cascaded.

RACF exit and DB2

DB2 exit routines can make significant changes in identification, authentication, access control and auditing. A new RACF exit, DSNXRAC, is supplied with DB2 9 for z/OS. Most of the information about these routines can be found in *DB2 Version 9.1 for z/OS Administration Guide*, SC18-9840. This book also documents tracing, instrumentation interfaces, and recovery log data used for audit.

The choice of using RACF for access control is not for everyone. There are significant policy and human resource implications. If you want the database administrators to manage security, integration with DB2 is important. If you want security administrators to manage security, integration with the security server is more important. As you make this change, roles will change and authorities will change. This is not a compatible change. You must plan to use RACF facilities more, like groups and patterns. The implementation team needs both DB2 and RACF knowledge for implementation.

If you want a security group to define authorization and a centralized security control point, this is a match for your needs. As you implement, plan to use patterns instead of individual item access authorities.

For more information about this topic, see the Roger Miller presentation *New DB2 Improvements in Security*, available at the following Web page:

<http://www.ibm.com/support/docview.wss?uid=swg27007843>

The DSNXRAC exit is a replacement for the RACF IRR@XACS exit that was supplied for use in earlier DB2 releases

The RACF access control module has changed substantially with DB2 9 for z/OS. It continues to integrate DB2 processing with RACF security, allowing for the consolidation of security administrative tasks and audit logs by moving the security management of DB2 objects and users into the RACF database.

It is not necessary to migrate protection of all DB2 objects at once. If the RACF access control module cannot find a RACF profile to protect a particular object, it defers to DB2 authority checking.

The relationship between the RACF resource classes and DB2 objects is listed in Table 3-1. For information about how implement DSNXRAC see IBM Redbooks publication *z/OS Version 1 Release 8 RACF Implementation*, SG24-7248.

Table 3-1 List of RACF resource classes

Class name	Description
DSNADM	DB2 administrative authority class
DSNR	Control access to DB2 subsystems
GDSNBP	Grouping class for DB2 buffer pool privileges
GDSNCL	Grouping class for DB2 collection privileges
GDSNDB	Grouping class for DB2 database privileges
GDSNJR	Grouping class for Java archive files (JARS)
GDSNPK	Grouping class for DB2 package privileges
GDSNPN	Grouping class for DB2 plan privileges
GDSNSC	Grouping class for DB2 schemas privileges
GDSNSG	Grouping class for DB2 storage group privileges
GDSNSM	Grouping class for DB2 system privileges
GDSNSP	Grouping class for DB2 stored procedures
GDSNSQ	Grouping class for DB2 sequences
GDSNTB	Grouping class for DB2 table, index or view privileges
GDSNTS	Grouping class for DB2 table space privileges
GDSNUF	Grouping class for DB2 user-defined function privileges
GDSNUT	Grouping class for DB2 user-defined distinct type privileges
MDSNBP	Member class for DB2 buffer pool privileges
MDSNCL	Member class for DB2 collection privileges
MDSNJR	Member class for Java archive files (JARS)
MDSNPK	Member class for DB2 package privileges
MDSNPN	Member class for DB2 plan privileges
MDSNSC	Member class for DB2 schema privileges
MDSNSG	Member class for DB2 storage group privileges
MDSNSM	Member class for DB2 system privileges
MDSNSP	Member class for DB2 stored procedures privileges
MDSNSQ	Member class for DB2 sequences
MDSNTB	Member class for DB2 table, index or view privileges
MDSNTS	Member class for DB2 table space privileges
MDSNUF	Member class for DB2 user-defined function privileges
MDSNDB	Member class for DB2 database privileges
MDSNUT	Member class for DB2 user-defined distinct type privileges

As an example, to allow USRT060 to select again from the EMP table, we now add a discrete profile with the SELECT privilege for EMP with universal access NONE to the MDSNTB class. Then we add a profile with READ access for USRT060 to the access list of this newly created discrete profile. We do this by submitting the job in Example 3-3.

Example 3-3 Giving USRT060 SELECT access on table EMP

```
//RUNDB2 EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
RDEF MDSNTB DB8L.DSN8810.EMP.SELECT UACC(NONE)
PERMIT DB8L.DSN8810.EMP.SELECT CLASS(MDSNTB) ID(USRT060) ACC(READ)
SETROPTS RACLIST(MDSNTB) REFRESH
/*
```

It is only necessary to activate one class and profile to activate the exit. However, adding another resource class needs a recycling of DB2. Updates to profiles, on the other hand, require a SETR RACLIST REFRESH of the class to activate.

Multilevel security in DB2

In multilevel security, the relationship between DB2 users and DB2 objects is important. In the context of multilevel security, a user is any entity that requires access to system resources.

Examples of users are as follows:

- ▶ Human resources
- ▶ Started tasks
- ▶ Batch jobs

In the context of multilevel security, an object is any system resource to which access must be controlled. Examples of objects are as follows:

- ▶ Data sets
- ▶ Tables
- ▶ Rows
- ▶ Commands

A user is an entity that requires access to system resources. Using multilevel security, you can define security for DB2 objects. By assigning security labels to your DB2 objects, you can define a hierarchy between those objects. Multilevel security then restricts access to an object based on the security level of that object.

DB2 also has an implementation of multilevel security at the row level. This implementation enables us to perform row-level security checks, which enable us to control which users have authorization to view, modify, or perform other actions on specific rows of data. For further information of how setup DB2 Multilevel security in DB2, see *Securing DB2 and Implementing MLS on z/OS*, SG24-6480.

Trusted contexts, roles, and MLS

A trusted context can exist without multilevel security. However, using them together allows for a user to be automatically switched to one of their defined SECLABELS for the duration of their work within the trusted connection. It also allows a default SECLABEL to be specified for the trusted context, thus associating a default SECLABEL to users that do not have one directly assigned (must also be a valid SECLABEL for those users). Combining the network trusted context, roles, and MLS allows more precise control of security.

3.2.3 SQL

SQL is generally separated into the ability to manipulate data. This means getting information through SELECT or modifying it through INSERT, UPDATE and DELETE. With SQL you can perform the following tasks:

- ▶ define data: CREATE new objects, ALTER them or DROP them
- ▶ data access and control: GRANT and REVOKE provide the built-in security.

There are many other interfaces into DB2: utilities, commands, and other Application Programming Interfaces (API). Security and authorization are included in GRANT and REVOKE for all of the interfaces.

Views

Views can perform the following tasks:

- ▶ Protect data: rows and columns
- ▶ Simplify access to data
- ▶ Join or union to add or remove information

Views can be used to hide data. They can provide only certain fields, as in Example 3-4.

Example 3-4 Using Data Definition Language

```
CREATE VIEW SW_CUSTOMER AS
SELECT CUST_NBR, CUST_NAME,
CUST_CREDIT
FROM CUSTOMER
WHERE CUST_REGION='SW'
```

Views are often used to simplify access to data by being able to define the view once and use it many times. Table privileges DELETE, INSERT, SELECT, UPDATE, or ALL can be granted individually on a view. By creating a view and granting privileges on it, you can provide access only to a specific combination of data. This capability is sometimes called field-level access control or field-level sensitivity.

Example 3-5 shows the ability to simplify. Using the view, only the additional qualification is needed. Often, a view can be used to handle more complex logic, such as a multiple table join or UNION (in Version 7). The person who uses the view does not need to be concerned with the join, UNION, or authorization concerns, if those are addressed in the view.

Example 3-5 Using a VIEW basic to standard

```
SELECT CUST_NBR, CUST_NAME,CUST_REGION
FROM SW_CUSTOMER
WHERE CUST_CREDIT = 'AAA'

SELECT CUST_NBR, CUST_NAME,CUST_REGION
FROM CUSTOMER
WHERE CUST_REGION = 'SW'
AND CUST_CREDIT = 'AAA'
```

You can use additional predicates to remove information without the view or retrieve from CUSTOMER table using the SW_CUSTOMER view.

Session variables

Session variables provide another way to provide information to applications. Some variables will be set by DB2. Others can be set in the connection and sign-on exits to set these session variables

A new built-in function GETVARIABLE is added to retrieve the values of a session variable. This function can be used in views, triggers, stored procedures and constraints to help enforce a security policy. If your primary security need is more general, this information complements other security mechanisms.

For example, you can have a view which provides data that is at the user's current security label:

```
CREATE VIEW V1 AS SELECT * FROM T1 WHERE  
COL5 = GETVARIABLE('SYSIBM.SECLABEL');
```

The session variables set by DB2 are qualified by SYSIBM. You can get the plan name, package name (schema, name and version), the user's seclabel, DB2 subsystem name, data sharing group name, version and CCSID information. This information is useful for security controls, but programmers have other needs for this information as well. Customers can add up to ten variables, with the qualifier SESSION, by setting the name and value in the connection and sign-on exits. Both the name and the value allow up to 128 characters. Session variables can be accessed, but not changed, in applications.

3.2.4 Application security

Applications do not have some of the protection mechanisms or the level of assurance provided by system security, so use the stronger system techniques whenever possible. Static SQL prevents a number of problems, including SQL injection, while improving performance. Static SQL authorization techniques can be used to avoid granting wide access to tables. If dynamic SQL is used, then use of parameter markers and host variables for input can also avoid SQL injection. Checking the input must be performed. Use of CONNECT with a password provides a shared technique and user ID that will make management more difficult.

Use system identification and authentication. Changing the password is needed more if you have passwords in programs.

There are several vendors for application security function.

3.2.5 Encryption

There are many ways to encrypt data in DB2. The questions, "What do you want to protect and from whom?" and "How much effort can be used?" are asked to determine which technique to use and where to encrypt and decrypt. Table 3-2 on page 69 summarizes these options.

Table 3-2 *Cryptography and DB2: options*

Where available	Characteristics
Outside of DB2 (ICSF, IBM Encryption for z/OS)	General, flexible, no relational range comparisons FOR BIT DATA
DB2 FIELDPROC	No relational range comparisons, FIELDPROC restrictions, FOR BIT DATA
DB2 EDITPROC (IBM tool)	indexes are not encrypted, EDITPROC restrictions
User-defined function or stored procedure	General, flexible, invocation needed, no relational range comparisons
SQL functions (DB2 V8)	General, flexible, invocation needed, no relational range comparisons
On the wire (DRDA V8, SSL V9, IPsec)	General, flexible
Tape Backup (z/OS, TS1120)	General, flexible, IBM hardware & software
Disk encryption	General, flexible, IBM hardware & software

Encryption does mean some trade-offs in function, usability, and performance. Either the indexes are not encrypted or encrypted data will not give correct results for comparisons other than equals or not equals. All greater than, less than, and range predicates are not usable. CPACF and CEX2 were designed to address high volume cryptographic transaction rates and bulk security requirements on z/OS. The Integrated Cryptographic Service Facility (ICSF) provides the interfaces to service routines supported by the hardware, such as key management. Products like the IBM Encryption Facility for z/OS and the IBM Data Encryption for IMS and DB2 Database Tool use the ICSF interface.

For performance considerations on encryption, refer to *DB2 UDB for z/OS Version 8 Performance Topics*, SG24-6465. DB2 V8 introduced a number of built-in functions which allow you to encrypt and decrypt data.

The encryption tool IBM Data Encryption for IMS and DB2 Database, is implemented using standard IMS exits and DB2 EDITPROCs. The exit or EDITPROC code invokes the System z Crypto Hardware to encrypt data for storage and decrypt data for application use, thereby protecting sensitive data residing on various storage media. This tool can save the time and effort required to write and maintain your own encryption software for use with such exits or within your applications.

The performance implications for encryption are roughly similar to data compression when only considering CPU overhead. The Data Encryption Tool is discussed in Part 5, "Data Encryption for IMS and DB2 Databases Tool" on page 275.

Data encryption has a number of challenges, including making changes to your application to encrypt and decrypt the data, encryption key management, and the performance overhead of encryption.

System z hardware has provided improving support for the encryption instructions and features, thereby decreasing the performance overhead of encryption. We mention the requirements when we discuss the functions used. The IBM System z cryptographic functions are described at the following Web page:

<http://www.ibm.com/servers/eserver/zseries/security/cryptography.html>

DB2 column level encryption

Starting with V8, DB2 for z/OS provides a number of built-in functions that allow you to encrypt data at the column level. These functions include ENCRYPT_TDES (or ENCRYPT) to encrypt data in a column, and DECRYPT_BIN and DECRYPT_CHAR to decrypt the data in its appropriate format, and the GETHINT function to retrieve the hint for the password.

Create and Insert

The SET ENCRYPTION PASSWORD statement allows you to specify a password as a key to encryption. In Example 3-6, the EMPNO in EMPL is encrypted with a password.

Example 3-6 DB2 data encryption

```
CREATE TABLE EMPL
(EMPNO VARCHAR(64) FOR BIT DATA,
EMPNAME CHAR(20),
CITY CHAR(20) NOT NULL DEFAULT 'KANSAS CITY',
SALARY DECIMAL(9,2))
IN DSND04.RAMATEST ;
COMMIT ;
SET ENCRYPTION PASSWORD = 'PEEKAY' WITH HINT 'ROTTIE' ;
INSERT INTO EMPL(EMPNO,EMPNAME, SALARY)
VALUES (ENCRYPT('12346'),'PAOLO BRUNI',20000.00) ;
INSERT INTO EMPL(EMPNO,EMPNAME, SALARY)
VALUES (ENCRYPT('12347'),'ERNIE MANCILL',20000.00) ;
```

When creating a column for data encryption, you must define it as VARCHAR. The length of the VARCHAR depends on the password and the password hint. Assuming EMPNO is VARCHAR(6) before encryption, you can compute the final length of VARCHAR as shown in Table 3-3.

Table 3-3 Example encrypted column VARCHAR size

Description	Bytes
Maximum length of non-encrypted data	6
Number of bytes to the next multiple of 8	2
Encryption key (24 bytes)	24
Optional password hint (32 bytes)	32
Total	64

Therefore, if you do not use a password hint, define the column for encrypted data as VARCHAR(32) FOR BIT DATA. If you use a password hint, DB2 requires an additional 32 bytes to store the hint. In this case, you must define the EMPNO column as VARCHAR(64) FOR BIT DATA.

You are responsible for managing all these keys. Make sure you have a mechanism in place to manage the passwords that are used to encrypt the data. Use the password hint. The GETHINT function returns the password hint for every row in the table.

```
SELECT GETHINT(EMPNO) FROM EMPL ;
-----+-----+-----+-----+
HINT
-----+-----+-----+-----+
ROTTIE
ROTTIE
DSNE610I NUMBER OF ROWS DISPLAYED IS 2
```

Without the password, there is no way to decrypt the data. These encryption functions use the Triple Data Encryption Standard (DES) to perform the encryption.

Select

To retrieve the data, the DECRYPT_CHAR function must be applied to EMPNO as in Example 3-7:

Example 3-7 Applying the DECRYPT_CHAR function

```
SET ENCRYPTION PASSWORD = 'PEEKAY' ;
SELECT SUBSTR(DECRYPT_CHAR(EMPNO),1,6) AS EMPNO,
EMPNAME,CITY,SALARY
FROM EMPL ;
```

This decrypts the EMPNO based on the password, as shown in Example 3-8:

Example 3-8 EMPNO decrypted

```
-----+-----+-----+-----+-----+-----+-----+
EMPNO EMPNAME      CITY      SALARY
-----+-----+-----+-----+-----+-----+-----+
12346 PAOLO BRUNI   KANSAS CITY 20000.00
12347 ERNIE MANCILL KANSAS CITY 20000.00
DSNE610I NUMBER OF ROWS DISPLAYED IS 2
```

Insert

Data is inserted into EMPL, and values for EMPNO are encrypted with the ENCRYPT function.

Because you can specify a different password for every row that you insert, you can encrypt data at the cell level in your tables as shown in Example 3-9:

Example 3-9 Encrypting data at the cell level

```
SET ENCRYPTION PASSWORD = 'ITSOSJ' WITH HINT 'SANJOSE' ;
INSERT INTO EMPL(EMPNO,EMPNAME, SALARY)
VALUES (ENCRYPT('12346'),'PAOLO BRUNI',20000.00) ;
SET ENCRYPTION PASSWORD = 'MANCILL' WITH HINT 'JACKSONVILLE' ;
INSERT INTO EMPL(EMPNO,EMPNAME, SALARY)
VALUES (ENCRYPT('12347'),'ERNIE MANCILL',20000.00) ;
```

In this case, the GETHINT function returns the information shown in Example 3-10:

Example 3-10 GETHINT information returned

```
SELECT GETHINT(EMPNO) FROM EMPL ;
-----+-----+-----+-----+
HINT
-----+-----+-----+-----+
SAN JOSE
JACKSONVILLE
DSNE610I NUMBER OF ROWS DISPLAYED IS 2
```

Prerequisites

The DB2 built-in encryption functions require the following prerequisites:

- ▶ DB2 V8.
- ▶ Integrated Cryptographic Service Facility (ICSF).
- ▶ On z890, z990 or later, CPACF is required (PCIXCC card is not, unless DRDA encryption is necessary).
- ▶ Pre-z890 and z990, cryptographic coprocessor is required.

Each CP on the z990 and later models has an assist processor on the chip in support of cryptography. This feature provides for hardware encryption and decryption support. PCIXCC provides a cryptographic environment with added function. To learn more about PCIXCC, refer to *IBM eServer zSeries 990 (z990) Cryptography Implementation*, SG24-7070.

Applications that need to implement DB2 encryption must apply the DB2 encrypt and decrypt built-in functions to each column to be encrypted or decrypted. All encrypted columns must be declared for bit data. Unchanged read-applications see data in encrypted form.

Applications may supply a different key for each column, but may also supply the key in a special register. We suggest, for performance, that you specify the key in the special register.

The LOAD and UNLOAD utilities do not support the DB2 built-in encryption functions, but do handle broader encryption. SQL-based programs such as DSNTIAUL support encryption. Encryption of numeric fields is not supported. The length of encrypted columns must allow for an additional 24 bytes, rounded up to a double-word boundary, for storing the encryption key. Space usage may be a concern if you plan to use DB2 to encrypt small columns. Indexes are also encrypted. Predicates that depend on the collating sequence of encrypted columns (for example, range predicates) may produce wrong results (unless modified to use built-in functions correctly).

For example, the following statement produces the wrong results.

```
SELECT COUNT(*) WHERE COL =:HV;
```

The following statement produces the correct results with almost no impact to performance.

```
SELECT COUNT(*) WHERE COL = ENCRYPT_TDES(:HV);
```

The following statement produces the wrong results.

```
SELECT COUNT(*) WHERE COL < ENCRYPT_TDES(:HV);
```

The following statement produces the correct results with a large impact on performance.

```
SELECT COUNT(*) WHERE DECRYPT_CHAR(COL) <:HV;
```

IBM Data Encryption for IMS and DB2 Databases

IBM Data Encryption for IMS and DB2 Databases is the tool that provides you with a data encryption function for both IMS and DB2 for z/OS databases in a single product. It enables you to protect your sensitive and private data for IMS at the segment level and for DB2 at the table level.

This tool performs encryption using EDITPROCs on the full row. Unlike the DB2 encryption functions shipped with DB2, the Data Encryption Tool uses different keys to encrypt different tables. The encryption keys can be either clear, such as the DB2 encryption functions, or secure. Plus they are managed through ICSF. Clear keys generally perform better. The tool also supports single, double, or triple DES. Refer to *IBM eServer zSeries 990 (z990) Cryptography Implementation*, SG24-7070, and *IBM System z10 Enterprise Class Technical Guide*, SG24-7516 to learn more about the clear and secure keys.

Customization consists of the following tasks:

- ▶ Building a user exit routine (DB2 EDITPROC exit)
- ▶ Putting the user-specified encryption key label in the exit routine

The tool provides sample jobs to help you build the user exit (DB2 EDITPROC) and specify the encryption key label. Alternatively, the ISPF window can be used to build the exit.

We provide an overview of the tool's function in 4.2, "Data Encryption for IMS and DB2 Databases Tool" on page 93 and Part 5, "Data Encryption for IMS and DB2 Databases Tool" on page 275.

The IBM Data Encryption for IMS and DB2 Databases Tool supports all versions of DB2. It encrypts only the whole data row. No application changes are required. However, you can include the EDITPROC only at the CREATE time of a table. DROP, CREATE, RUNSTATS, and BIND are necessary for adding the EDITPROC to an existing table. The applications do not need to be aware of encryption keys.

The tool takes care of data encryption on disk (data at rest). The data on channel, the buffer pools, the image copies, and the logs are encrypted. Data passed to applications and the indexes are not encrypted. This means that existing authorization controls are unaffected. Once the data is brought, for example, into DBMS working storage, you are using the existing DB2 and IMS authorizations to secure data.

In the case of two data sharing members on two processors, they both must use the same encryption key.

Recommendations

For both DB2 encryption and the IBM Data Encryption Tool, we suggest you move to a more current System z hardware, where the hardware-assisted encryption instructions are available on all processors. The IBM Encryption Tool (row level) generally performs better than the DB2 encryption and decryption (column level). The break even point varies depending on how many other columns are in the table and their size.

However, performance is not the only reason to choose one encryption strategy over the other, as they also vary in function. If the data is defined as compressed, because encryption is done before compression, compression and encryption cannot be effectively combined. Encrypted data does not tend to compress well because repeating unencrypted characters are no longer repeating after they are encrypted. A new option in the IBM Encryption Tool allows compression before encryption within the EDITPROC. See 14.3, "Compression and encryption" on page 334.

3.2.6 Network security

Network security is a important area. The Kerberos and Passticket techniques provide the best security for identification and authentication. If the data is sensitive and uses a network, then protecting or encrypting the network communication is required. If you need to use passwords, then encrypt SYSIBM.SYSUSERNAMES.

APAR PQ95205 for V8 adds the ability to encrypt the user and password in the SYSIBM.USERNAMES catalog table.

This is a summary of items to consider:

- ▶ Kerberos, Passticket
- ▶ Enterprise Identity Manager
- ▶ Encryption of passwords
- ▶ Encryption on Web and on disk
- ▶ Encrypt session
- ▶ DRDA Standard
- ▶ SSL
- ▶ Do NOT use already verified
- ▶ Do not trust a client

In any case, if you use already verified or trust a client, then the server can be compromised whenever a client is compromised.

These are generally unacceptable choices with today's understanding of security.

See *DB2 Version 9.1 for z/OS Administration Guide, SC18-9840* for more on this topic.

3.2.7 Auditing

Do you monitor your database? If not, then you will even notice when there are breaches of security. Unfortunately, no security system is perfect. The major losses often start simply, when someone accidentally discovers that they have access that should not have been defined.

Not monitoring or not auditing for security breaches is not adequate in any situation where the value of the data exceeds the time to monitor. Even basic, low security, or minimal practice requires audit for security failures. Higher levels of security require more auditing.

There are many kinds of audit information available in DB2. The DB2 catalog stores the definitions of all the objects and the authorization. Users allowed to access these tables can use the power of SQL to audit and manage security. The DB2 audit facility integrated into z/OS can be turned on to track user actions in the DB2 database system. RACF has an unload facility that allows its security definitions to be loaded into DB2, so that broader auditing can be performed.

The DB2 recovery log and utilities or tools are also helpful in finding out how and when some data was modified.

You can also use external functions such as the tool IBM DB2 Audit Management Expert for z/OS.

DB2 auditing trace

DB2 for z/OS Instrumentation Facility Component supports not only accounting data of applications but also audit data for applications.

Table 3-4 on page 75 summarizes the audit classes. For details about audit class, see *DB2 Version 9.1 for z/OS Administration Guide, SC18-9840*.

Table 3-4 Description of audit classes

Audit class	Activated IFCIDs	Events that are traced
1	140	Access attempts denied due to inadequate authorization
2	141	Explicit GRANT and REVOKE
3	142	CREATE, ALTER, and DROP operations against audited tables
4	143	First change of audited object
5	144	First read of audited object
6	145	Bind time information about SQL statements that involve audited objects
7	55, 83, 87, 169, 319	Assignment or change of authorization ID
8	23, 24, 25, 219, 220	Utilities
9	146	Various types of records written to IFCID 146 by the IFI WRITE function
10	269, 270	CREATE and ALTER TRUSTED CONTEXT statements, establish trusted connection information and switch user information

If more detailed information is required, examine the individual SQL statements (IFCID 0350), input host variables (IFCID 0247), record accesses and locking. Other detailed trace data allows full accounting by user or transaction, with detailed data written every plan deallocation or change of user and fully detailed tracing down to individual call / IO / component level.

Collecting audit data

To collect audit data, you need to start audit trace. From DB2 9 for z/OS, the newly introduced object ROLE can be used to filter audit trace. Example 3-11 shows how to start audit trace with including or excluding specific role.

Example 3-11 Start trace with including or excluding roles

```
-- Start audit trace with including role
START TRACE(AUDIT) CLASS(...) ROLE(...)

-- Start audit trace with excluding role
START TRACE(AUDIT) CLASS(...) XROLE(...)
```

You also need to enable tables to be auditable. To turn on audit for the existing table, you can use the ALTER TABLE statement in Example 3-12.

Example 3-12 Alter table for auditing

```
ALTER TABLE <table_to_be_audited> AUDIT ALL/CHANGES
```

Reporting audit data

You can report audit data by using OMEGAMON® PE. Example 3-13 on page 76 shows the usage of OMEGAMON PE for audit trace.

Example 3-13 Using OMEGAMON PE for audit trace

```
//DB2PE EXEC PGM=DB2PE
//INPUTDD DD DSN=SMFDATA,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//JOBSUMDD DD SYSOUT=*
//AUDTRC DD SYSOUT=*
//SYSIN DD *
AUDIT TRACE DDNAME(AUDTRC)
INCLUDE(CONNECT(SERVER))
EXEC
```

We provide a sample audit trace along with the steps of the test JDBC application. The test JDBC application performs the following tasks:

- ▶ Connects to database DB9A using AUTHID PAOLOR1 from 9.189.188.76. This connection is classified into TRUSTED CONTEXT SG247720 associated with the ROLE ITSOBANK.
- ▶ Issues a DELETE statement. This is the first write to the audited table COMPANY_A.
- ▶ Issues a INSERT statement.

Example 3-14 shows the audit trace from the test JDBC application as edited by OMEGAMON PE.

Example 3-14 Audit trace sample output

OPRMAUTH	CORRNAME	CONNTYPE	ORIGAUTH	CORRNMBR	INSTANCE	PLANNAME	CONNECT	TIMESTAMP	TYPE	DETAIL
PAOLOR1	db2jcc_a	DRDA	PAOLOR1	ppli	C350F798E892			23:38:21.06	AUTHCHG	TYPE: ESTABLISH TRUSTED CONTEXT OBJECT OWNER: AUTHID SECURITY LABEL: CONTEXT NAME: SG247720 CONTEXT ROLE: ITSOBANK USER ROLE: PREV. SYSAUTHID: PAOLOR1 REUSE AUTHID: SERVAUTH NAME: JOB NAME: ENCRYPTION: NONE TCP/IP USED: 9.189.188.76
										STATUS: SUCCESS SQLCODE: 0
PAOLOR1	db2jcc_a	DRDA	PAOLOR1	ppli	C350F798E892			23:38:21.43	BIND	PACKAGE: DB9A.NULLID.SYSLH200.X'5359534C564C3031'
										TYPE: DELETE STMT#/P ISOLATION(CS) KEEP UPD LOCKS: NO TEXT: DELETE FROM COMPANY_A DATABASE: DSN00293 TABLE OBID: 3
PAOLOR1	db2jcc_a	DRDA	PAOLOR1	ppli	C350F798E892			23:38:21.44	DML	TYPE : 1ST WRITE DATABASE: DSN00293 PAGESET : COMPANYR
										TABLE OBID: 3 LOG RBA : X'0000000000000'
PAOLOR1	db2jcc_a	DRDA	PAOLOR1	ppli	C350F798E892			23:38:21.78	BIND	PACKAGE: DB9A.NULLID.SYSLH200.X'5359534C564C3031'
										TYPE: INSERT STMT#/P ISOLATION(CS) KEEP UPD LOCKS: NO TEXT: INSERT INTO COMPANY_A VALUES(5791, 'O''Brien', 38, 'Sales', NULL, 18006.00) DATABASE: DSN00293 TABLE OBID: 3

Audit trace filtering and overhead

New filtering capabilities for `–START TRACE` that `INCLUDE` or `EXCLUDE` based on these keywords have been provided with DB2 9 for z/OS:

- ▶ `USERID`: Client user ID
- ▶ `WRKSTN`: Client workstation name
- ▶ `APPNAME`: Client application name
- ▶ `PKGLOC`: Package `LOCATION` name
- ▶ `PKGCOL`: Package `COLLECTION` name
- ▶ `PKGPROG`: `PACKAGE` name
- ▶ `CONNID`: Connection ID
- ▶ `CORRID`: Correlation ID
- ▶ `ROLE`: Database `ROLE`

Improved trace filtering makes the jobs of auditing and of performance management easier. Many more options can minimize the amount of data collected, so the overhead is reduced and the extraneous data does not need to be processed.

In V7 time-frame, we measured 1200 instructions for accessing or updating an audited table. In V8 CPU time measurement, we observed 7.2% overhead for turning on all audit trace classes for one online transaction. A typical overhead for an online transaction is expected to be less than 10% even when all audit trace classes are turned on. An overhead as high as 10% or more is possible for transactions that execute many short-running and distinct SQL statements just once per transaction, because an audit trace overhead is encountered only the first time an audited table is accessed or updated in a transaction.

For utility, batch, and long-running SQL, the overhead is practically zero.

IBM DB2 Audit Management Expert for z/OS

IBM DB2 Audit Management Expert for z/OS collect log and trace data in an audit repository where auditors are able to then view, analyze, and generate comprehensive reports on the data using selectively filtered `SELECT`, `INSERT`, `UPDATE`, and `DELETE` activity by user or by object. We have an overview at 4.1, “DB2 Audit Management Expert for z/OS” on page 92 and details on using this tool in Part 4, “DB2 Audit Management Expert” on page 161.

DB2 WebSphere auditing

If we make the assumption that the security mechanism for authentication will result in all of the users for an application exploiting the same user ID and password, the ability for the application to perform auditing tasks is significantly reduced. To help with this issue, DB2 provides “client strings” that can be set by the application, which can help applications provide specific user information to the client. An application may choose to set these fields to the user of the application if the data was propagated back to the data access portion of the application.

The client strings are actually four different strings (Table 3-5 on page 78) that can be set to any value of your choosing. The value of these strings shows up in the output of the `–DISPLAY THREAD` command and DB2 trace records. Those can be processed by batch reporting tools or online monitoring tools such as DB2 Performance Expert. Refer to your reporting user’s guide to generate reports including these strings, or how to limit results to a specific value.

Table 3-5 Client accounting strings

String	Maximum length	In standard header
ClientWorkstation	18	YES
ClientUser	16	YES
ClientProgramName	12	YES
ClientAccountingInformation	22	NO

3.3 IMS

This section discusses how IMS for z/OS implement the security capabilities.

IMS was developed prior to the introduction of RACF. As a result, it initially incorporated its own security mechanisms to control user access to the various IMS resources, transactions, databases, programs, and so forth. This security was controlled by a number of means. A number of security exits were provided. Also, a series of bitmaps defined users and their access to resources. This is referred to as a security matrix. These are load modules, produced by the IMS security maintenance utility following the generation of an IMS system.

With the introduction of RACF, IMS has been enhanced to make use of RACF for controlling access to IMS resources. It is possible to use the original IMS security features, the new RACF features, and combinations of these. Using RACF provides more flexibility than the older security features.

The normal features of RACF can be used to protect both system and database IMS data sets.

Two advantages of using a security product for securing access to resources are as follows:

- ▶ One product may be used to implement the security requirements for multiple subsystems, such as IMS, CICS, and other subsystems.
- ▶ All of the security information may be kept and maintained in one place, like the RACF database. One centralized database repository containing all the installations security specifications eliminated, or significantly minimized, the previous requirements to have the following circumstances in effect:
 - Security information distributed among several subsystems.
 - The security enforcement functions implemented in multiple products RACF offered a wide range of security choices to the installation.

For example, RACF contains security features, such as user identification (user ID) and verification-based security that is not available through IMS internally provided SMU security.

3.3.1 Authorization

The SECURITY macro was implemented in IMS mainly for the purpose of specifying the installations security choices to an external security product, like RACF. Again, IMS also provided keywords and parameters on the SECURITY macro that allowed security specifications for SMU-provided security and for installation-provided security exits. That is,

the same security options that could be specified on the COMM and IMSGEN macros could now be specified on the SECURITY macro. This maintained the compatibility between security specifications on the SECURITY, COMM, and IMSGEN macros.

Protecting IMS terminals

Two types of terminals may be protected in the IMS environment:

- ▶ Physical terminals
- ▶ Logical terminals or LTERMS.

The physical terminals may be static terminals (those defined to IMS using the TYPE and TERMINAL macros); or Extended Terminal Option (ETO) terminals that are dynamically defined to IMS at sign-on.

Terminals can be protected for the following circumstances:

- ▶ Sign-on verification, which will determine whether sign-on is required
- ▶ Resource access security, which will determine whether the terminal has access to issue-specific IMS commands, or transactions.

Protecting IMS commands

IMS commands request the system to perform specific functions, such as displaying information about IMS resources or altering the status of system resources. IMS commands may be entered from several sources:

- ▶ A user terminal.
- ▶ The IMS master terminal.
- ▶ Multiple Console Support/Extended Multiple Console Support (MCS/EMCS) MVS consoles.
- ▶ Automated operator programs that issue the DL/I CMD call or the DL/I ICMD call.

The security facility used to determine whether or not the command will be processed depends on the origin of the command. For example, the command may have been entered from a static or ETO terminal, or from a program that issued either the DL/I CMD or ICMD call. Furthermore, DFSCCMD0 may also be customized by the installation to provide a more granular level of security. This exit can be used in conjunction with SMU or RACF, or it may be used alone without SMU or RACF to provide command security.

Protecting IMS transactions

There are six methods used to secure IMS transactions:

- ▶ Resource access security (LTERM based)
As with IMS LTERM-based command security, SMU is used to determine which LTERMS may be used for entering transaction codes. As previously mentioned, password and LTERM-based security may be combined. Because SMU is used to provide this type of security, the physical terminal (that is associated with the LTERM) must be statically defined to IMS.
- ▶ Resource access security (password based)
Transaction authorization based on securing the transaction with a password is implemented using SMU. Thus, the terminals from which the password-protected transactions are entered are required to be static terminals.

- ▶ Extended resource access security

If SMU-provided LTERM-based and password security is insufficient to meet the installation requirements, the Transaction Authorization Exit (DFSCTRNO) routine may be customized to meet the requirements.
- ▶ Resource access security (user ID based)

RACF enforces transaction authorization security by checking the TIMS/GIMS RACF classes for transaction security profiles. If a security profile for a transaction exists in one of the classes (TIMS or GIMS) the transaction has been secured and protected. RACF checks to see if the user ID (or group name) has been authorized to execute the transaction.
- ▶ Extended resource access security (user ID based)

If RACF provided transaction authorization security is insufficient to meet the installation requirements, the DFSCTRNO routine may be customized to meet the requirements. RACF would be called first to determine if the user ID/group name is permitted to execute the transaction. On successful authorization by RACF, DFSCTRNO would be called to perform installation-specified transaction authorization security checking.
- ▶ User customizable (DFSCTRNO)

This exit is enabled within the IMS gen SECURITY macro, and can be tailored to suit any other type of security required.

For more information, see *IMS V6 Security Guide*, SG24-5363.

3.3.2 Encryption

In IMS, the IBM Data Encryption tool implements encryption using the standard segment edit/compression exit routine. Both IMS data and index databases can be encrypted and decrypted.

You can write a segment edit/compression exit routine to encoding and decoding segments for security purposes. The logic for data encoding and decoding can be based on information contained within the user-written routine itself. It also can be based on information from an external source, such as data provided in the DBD block, or from tables examined at execution time. The segment edit/compression exit routine is optional. No default routine is called. These routines should be transparent to the application programs that access the databases.

Restriction

When you specify the maximum size of the data portion of the segment in the DBD, and you use the segment edit/compression exit routine with full-function variable-length segments, you might need to include extra bytes. These extra bytes are needed if your exit routine makes the segment larger than its maximum size. For example, if the maximum length of your data is 100 bytes and your exit routine might add 2 bytes to the segment, specify 102 bytes as the maximum size. Increasing the maximum size accounts for the size of the segment from the application program (100 bytes) and the 2 bytes added by the exit routine. This restriction does not apply to full function fixed-length segments or to segments in a data entry database (DEDB). Using the segment edit/compression exit routine for both types of segments might increase their data sizes to values that are larger than those specified in the DBD. The attributes of the segment edit/compression exit routine are listed in Table 3-6 on page 81.

Table 3-6 Attributes of the segment edit/compression exit routine

Attribute	Description
IMS environments	All environments that support databases
Naming convention	According to user's naming convention
Link editing	After an edit routine has been compiled and tested and before it is used by the IMS system, it must be placed into IMS.SDFSRESL, SYS1LINKLIB, or into any operating system partitioned data set to which access is provided with JOBLIB or STEPLIB control region JCL statement. You must also specify one entry point to the exit routine
Including the routine	Routine is specified in the SEGM macro for DBDGEN
IMS callable services	To use IMS callable services with this routine, you must do the following: <ul style="list-style-type: none"> ▶ Issue an initialization call (DFSCSI0) to obtain the callable service token and a parameter list in which to build the function-specific parameter list for the desired callable service. ▶ Use the PST address found in register 1 as the ECB. ▶ Link DFSCS100 with your user exit.
Sample Routine location	IMS.ADFSSRC

Loading the routine

Each time a database is opened, IMS examines each segment description to determine whether edit/compression has been specified for that segment type. If so, the exit routine is loaded from its resident library by IMS. IMS obtains the name of the routine from the COMPRTN= parameter of the SEGM statement of the DBD.

An IMS restart is required to refresh the loaded exit routine with a new version.

How the segment edit/compression facility works

When a segment requiring editing, encoding, decoding, or compression is accessed, IMS gives your edit routine control and provides it with the following information:

- ▶ Address of the data portion of the segment
- ▶ Address of the segment work area

Although your edit routine can modify the key fields in a segment, the segment's position in the database is determined by the original key field. For example, if the key field of a segment type is based on last names and the database has segments for people named McIvor, Hurd, and Caldwell, these segments are maintained in alphabetic sequence-Caldwell, Hurd, and McIvor. Assume your edit routine encodes the names as follows:

```
Caldwell -----> 29665
Hurd -----> 16552
McIvor -----> 24938
```

The encoded value is put in the key field. However, the segments in the database remain in their original sequence (Caldwell, Hurd, McIvor) rather than in the numeric sequence of the encoded values (16552, 24938, 29665). Because segments in the database are maintained in their original sequence, application programs can issue GN calls and retrieve the correct segment even though segments are encoded. This is also true for secondary index fields contained in index source segments.

Restrictions

Keep the following restrictions in mind when using the segment edit/compression facility:

- ▶ Because this routine becomes a part of the IMS control or batch region, any abnormal termination of this routine terminates the entire IMS region.
- ▶ The exit routine cannot use operating system macros such as LOAD, GETMAIN, SPIE, or STAE.
- ▶ All encoding of segments occurs as the segments are described in a physical database only.

The exit routine must not modify or alter the relative position of a key field in a DEDB segment. If the key field in a DEDB segment changes or moves during a compress or expand call, IMS issues abend 0799, subcode 1.

3.3.3 Auditing

IMS Audit Management Expert provides the following features and functions:

▶ Data collection

IMS Audit Management Expert can collect and correlate many different types of information into its audit repository:

- Access to database data sets and image copy data sets as recorded in SMF
- Access to databases as recorded in the IMS log
- User access to the IMS system through SIGNON as recorded in the IMS log
- PSB and database change of state activity as recorded in the IMS log
- System STOP and START activity as recorded in the IMS log.

▶ Reporting user interface

Provides auditors with flexible options for examining the data in the audit repository.

▶ Administration user interface

Provides administrators with flexible options for user management, auditing profiles, and reporting authorizations.

IMS Audit Management Expert components

The IMS Audit Management Expert for z/OS agent communicates with the IMS Audit Management Expert for z/OS server and data repository.

Agent

The agent is responsible for collecting and filtering data based on predefined criteria. Each agent collects data from a single IMS. The agent populates the repository with event types from the IMS Log and SMF.

Data sharing support

A common (shared) Database Recovery Control (DBRC) RECON data set controls concurrent access to databases by IMS systems in one or more operating systems. Databases that are intended to take part in data sharing must be registered in the RECON. Each registered database has a current status that reflects whether it can take part in data sharing. The IMS collector uses the RECON to discover log data sets. If the RECON and the logs can be accessed by the IMS collector, audit data will be collected. This means IMS Audit Management Expert for z/OS supports IMS data sharing.

Data collectors

IMS Audit Management Expert for z/OS provides users with a number of options to audit their IMS environment.

Information is collected primarily from the RECON, IMS, and SMF logs. Only databases registered with DBRC can be audited by IMS Audit Management Expert. A DB2 repository is used to save the collected information. The user is able to view the information by means of a client user interface.

Audit repository

The IMS Audit Management Expert for z/OS repository stores the audit data collected by the agent. The agent captures audit data and stores it in a set of DB2 tables referred to as the audit repository. The DB2 audit repository contains tables to define MVS systems and their properties. Each audit repository is associated with an IMS Audit Management Expert for z/OS server and is stand-alone. Multiple audit repositories within the IMS Audit Management Expert environment are permitted.

Audit server

The audit server is the central control point for an IMS Audit Management Expert for z/OS network. The server transmits control data from the audit repository to the agents. A single audit server can support data collection from multiple agents on multiple MVS systems, including multiple IMS instances. The audit server performs the following functions:

- ▶ Supports the administration user interface
- ▶ Supports the reporting user interface
- ▶ Accesses the audit repository
- ▶ Sets up monitoring criteria

Administration user interface

Provides administrators with flexible options for user management, auditing profiles, and reporting authorizations.

User interfaces IMS Audit Management Expert for z/OS provides two user interfaces:

- ▶ Administration user interface
This interface enables IMS Audit Management Expert for z/OS product administrators to perform administrative tasks such as creating collections and collection profiles, and adding and modifying user profiles.
- ▶ Reporting user interface
This interface enables all IMS Audit Management Expert for z/OS users to view audit data and reports collected by IMS Audit Management Expert for z/OS.

IMS Audit Management Expert collected resources

IMS Audit Management Expert for z/OS provides the ability to collect and correlate data access information from a variety of IMS resources such as RECON, IMS, and SMF logs. See Table 3-7 on page 84.

Table 3-7 IMS log types collected by IMS Audit Management Expert

Log type number	IMS log type	IMS log type description	IMS collector event
06	IMS/VS accounting record x'06'	IMS online was started or stopped. IMS batch was started or stopped. IMS stopped for planned RSR takeover.	IMS006
07	Application terminate accounting log record	An application program terminated	End IMS050
08	Application start accounting log record	An application program was scheduled	Start IMS050
0A07	A CPI communications driven application program was terminated	A CPI communications driven application program was terminated	End IMS050
0A08	A CPI communications driven application program was started	A CPI communications driven application program was started	StartIMS050
16	A /SIGN command successfully completed	A /SIGN command successfully completed	IMS016
20	A database was opened	A database was opened	IMS020
21	A database was closed	A database was closed	IMS020
4C	DB/PSB Activity	Activity related to database or PSB processing	IMS04C
50	Database update	The database was updated. This log record contains the new data on an insert and update call and the old data and FSE updates on a delete call	IMS050
5921	DEDB ADS OPEN/CLOSE/STATUS log record	DEDB area data set was opened	IMS020
5922	DEDB ADS OPEN/CLOSE/STATUS log record	DEDB area data set status was closed	IMS02C
5923	DEDB ADS OPEN/CLOSE/STATUS log record	DEDB area data set status was changed	IMS04C
595_	FP DB record was changed	FP DB record was changed	IMS050

SMF is used to obtain additional data set activity related to the monitored IMS databases and image copies. See Table 3-8.

Table 3-8 SMF record type and contexts

Record Number	Record subtype	SMF context
14		OPEN read
15		OPEN update
17		Deleted
18		Renamed
30		Accounting
60	Insert	Created
62	Update	OPEN update
64		CLOSE
65		Deleted
66		ALTER
66	Rename	Rename

Restrictions

Only databases registered with DBRC can be audited by IMS Audit Management Expert

IMS Audit Management Expert can only report events that are being collected by SMF. If any SMF record type in Table 3-8 is not being collected, then IMS Audit Management Expert cannot report that event.

3.4 VSAM

The security facilities provided by VSAM generally are not sufficient for an installation that is serious about data security. For one thing, too many passwords are involved. With VSAM, it is possible to assign as many as twelve (12) different passwords to each VSAM KSDS (four each for the cluster, the data component, and the index component). As a result, a shop can end up managing thousands of different passwords.

Because of the inherent weaknesses of VSAM security, most installations use other security managers that provide a more comprehensive, system-wide approach to security such as RACF.

One final point about the RACF product is that RACF completely replaces the password protection feature under VSAM. As a result, when a data set is RACF-protected, any VSAM password specification coded is ignored.

3.4.1 Authorization

RACF can protect your VSAM data sets from other users by controlling who has authority to access them and at what authority level they can do so. You can use RACF to protect data sets by creating profiles for them. When you attempt to use a data set, RACF checks your user profile and the data set profile to decide whether to allow you to use it. A data set profile contains the following information:

- ▶ The data set name
- ▶ The data set owner
- ▶ The access list, which is a list of specific users and groups who can use a data set and how they can use it
- ▶ The universal access authority (UACC), which is the default level of access authority allowed for all users or groups not specified in the access list.
- ▶ Auditing information

You can protect a data set by identifying specific users or groups with the access you want them to have in the access list. You can give all other RACF-defined users a certain access. Just put ID(*) in the access list with the access authority you want them to have. All other users are allowed the access you specify as the universal access authority (UACC). The access authorities you can specify are: NONE, READ, UPDATE, CONTROL, ALTER, and EXECUTE.

- ▶ ALTER allows users to read, update, or delete the data set.
- ▶ CONTROL provides users with the same authority that is provided with the VSAM CONTROL password: Authority to perform control interval access (access to individual VSAM data blocks) and to retrieve, update, insert, or delete records in the specified data set.
- ▶ UPDATE allows users to read or update the data set. UPDATE does not, however, authorize a user to delete the data set.
- ▶ READ allows users to access the data set for reading or copying only.
- ▶ NONE does not allow users to access the data set.

Attention: Anyone who has READ, UPDATE, CONTROL, or ALTER authority to a protected data set can create a copy of it. As owner of the copied data set, that user has control of the security characteristics of the copied data set, and can downgrade it. For this reason, you might want to initially assign a UACC of NONE, and selectively permit a small number of users to access your data set, as their needs become known

3.4.2 Encryption

In compatibility mode, ICSF supports the Access Method Services Cryptographic Option

The Access Method Services user can use REPRO to encipher data that is written to a data set, and then store the enciphered data set offline. When desired, you can bring the enciphered data set back online, and use REPRO to decipher the enciphered data. You can decipher the data either on the host processor on which it was enciphered, or on another host processor that contains the Access Method Services Cryptographic Option and the same cryptographic key that was used to encipher the data. You can either use ICSF to create the cryptographic keys, or use keys that the Access Method Services user supplies.

With the exception of catalogs, all data set organizations that are supported for input by REPRO are eligible as input for enciphering. Similarly, with the exception of catalogs, all data set organizations supported for output by REPRO are eligible as output for deciphering. The resulting enciphered data sets are always sequentially organized (SAM or VSAM entry-sequenced data sets).

Example 3-15 shows an IDCAMS REPRO using a clear key to encipher.

Example 3-15 Encipher using a Clear Key in REPRO command

```
//ENC EXEC PGM=IDCAMS,REGION=4M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
REPRO -
  INDATASET('PAOLOR9.CLEAR') -
  OUTDATASET('PAOLOR9.CRIPT') -
  ENCIPHER(PRIVATEKEY DATAKEYVALUE('01020102'))
```

Example 3-16 shows the output from the encipher job.

Example 3-16 Report from encipher JCL

```
IDCAMS SYSTEM SERVICES
REPRO -
  INDATASET('PAOLOR9.CLEAR') -
  OUTDATASET('PAOLOR9.CRIPT') -
  ENCIPHER(PRIVATEKEY
  DATAKEYVALUE('01020102'))
IDC0005I NUMBER OF RECORDS PROCESSED WAS 9
IDC0001I FUNCTION COMPLETED, HIGHEST
CONDITION CODE WAS 0
IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM
CONDITION CODE WAS 0
```

Example 3-17 shows an IDCAMS REPRO using a clear key to decipher.

Example 3-17 Decipher using a Clear Key in REPRO command

```
//ENC EXEC PGM=IDCAMS,REGION=4M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  INDATASET('PAOLOR9.CRIPT') -
  OUTDATASET('PAOLOR9.CLEAR.AGAIN') -

  DECIPHER(DATAKEYVALUE('01020102'))
```

Example 3-18 on page 88 shows an IDCAMS REPRO using an enciphered key to encipher.

Have IDCAMS generate a data key that will be protected by KEK (EXPORTER) that is shared with communication party stored in CKDS with name KEKREPEX printed in the IDCAMS report and data in its encrypted form (X'0969..).

Example 3-18 Encipher using an encrypted key

```
//ENC EXEC PGM=IDCAMS,REGION=4M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
REPRO -
INDATASET('PAOLOR9.CLEAR') -
OUTDATASET('PAOLOR9.CRIPT') -
ENCIPHER -
(EXTERNALKEYNAME(KEKREPEX))
```

Example 3-19 shows an IDCAMS REPRO using an enciphered key to decipher.

Decipher using an encrypted key; KEK (EXPORTER) must be stored in cryptographic key data set on receiving system with name KEKREPEX Use encrypted key as printed in the IDCAMS report (X'0969..)

Example 3-19 Decipher using an encrypted key

```
//DEC EXEC PGM=IDCAMS,REGION=4M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
REPRO -
INDATASET('PAOLOR9.CRIPT') -
OUTDATASET('PAOLOR9.CLEAR.AGAIN') -
DECIPHER -
(SYSTEMKEY -
SYSTEMDATAKEY(X'0969D3115EA7F2B4') -
SYSTEMKEYNAME(KEKREPEX))
```

The report from the decipher job is listed in Example 3-20.

Example 3-20 Output of decipher using an encrypted key

```
IDCAMS SYSTEM SERVICES
REPRO -
INDATASET('PAOLOR9.CRIPT') -
OUTDATASET('PAOLOR9.CLEAR.AGAIN') -
DECIPHER -
(SYSTEMKEY -
SYSTEMDATAKEY(X'0969D3115EA7F2B4') -
SYSTEMKEYNAME(KEKREPEX))
IDC0005I NUMBER OF RECORDS PROCESSED WAS 7
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION
CODE WAS 0
IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM
CONDITION CODE WAS 0
```

The REPRO command's encryption algorithm variables are not documented, so you cannot use them to write decryption applications on another system. Therefore, cross-platform exchange is not possible.

Restriction: AMS REPRO Encryption is not supported when running FMID HCR7708 on an IBM zSeries® 990.

3.4.3 VSAM auditing

RACF logs data set utilization on SMF to provide auditing records. The SMF records are listed in Table 3-9.

Table 3-9 SMF Records

Type	Description
60	VSAM Volume Data Set Updated
62	VSAM Component or Cluster Opened
63	VSAM Catalog Entry Defined
64	VSAM Component or Cluster Status
67	VSAM Catalog Entry Delete
68	VSAM Catalog Entry renamed
69	VSAM Data Space, defined, extended or deleted
80	RACF Processing
81	RACF Initialization
83	RACF Processing Record for Auditing Data Sets

The auditing granularity can be the use of specific data set or general. For example, you can define the profile to audit only the attempts to update on a specific data set, or, you can set up a resource profile for your data set to audit every attempt to use a data set.

There are several ways to look into these SMF records:

- ▶ IBM SystemView® Enterprise Performance Data Manager/MVS (EPDM) program
- ▶ The DBU2MSXL, a set of scripts that loads the output of the RACF Database Unload Utility (IRRDBU00) into Microsoft® Excel® spreadsheet.
- ▶ The DBU2MSAC, a set of scripts that loads the output of IRRDBU00 into Microsoft Access
- ▶ DFSORT ICETOOL utility can analyze and produce reports from IRRDBU00 output.

The tools DBU2MSXL and DBU2MSAC can be downloaded from the following Web page:

<http://www.ibm.com/servers/eserver/zseries/zos/racf/downloads>

For more information, see *OS/390 Security Server Audit Tool and Report Application*, SG24-4820.



IBM information management tools

Sarbanes-Oxley and many other regulations have placed a heavy technical burden on today's IT staff to fulfill auditors' requests for various views and reports of data. Auditors now need to track, analyze, and report on the status of legal and regulatory compliance efforts. Until now, there has been no easy way for auditors to access the data they need, other than relying on the IT department.

IBM provides a rich set of tools to provide auditors, security administrators, and database administrators (DBAs) the capabilities they need to deliver accurate, timely data and reports for use in auditing activities.

In this chapter we provide a high-level discussion of the IBM tools that address the auditing and compliance needs for auditors and security administrators, in addition to the security functions a DBA typically requires.

The tools mentioned in this chapter are as follows:

- ▶ "DB2 Audit Management Expert for z/OS" on page 92
- ▶ "Data Encryption for IMS and DB2 Databases Tool" on page 93
- ▶ "Log Analysis Tool" on page 95
- ▶ "Performance tools" on page 95

4.1 DB2 Audit Management Expert for z/OS

Today's auditors want the knowledge of who, what, when, where, why, and how data is accessed. This is a difficult and time consuming effort. It requires pulling together all the information required in an audit, but they mostly have to rely on technical personnel like application developers and DBAs to collect and report information. This approach has many drawbacks:

- ▶ Collection of audit data

Existing developer and DBA tools are not audit-oriented and are not designed to collect all of the relevant audit information from the source.

- ▶ Reporting of audit data

Existing developer and DBA tools are not audit-oriented, nor are they designed to present information in a useful way from an auditor's perspective.

- ▶ Integrity of all audit information

Because DBAs are part of the audited population, the auditor should not be dependent on them to provide key audit information.

In addition, DBA user identifications (user IDs) mostly have more system-level privileges than typical business users, which gives them more opportunity to circumvent normal business controls

The DB2 Audit Management Expert for z/OS is an auditing solution that enables customers to segregate duties in an easy, centralized environment. It provides complete automation of the auditing processes, which reduces fraud exposures, and the high costs associated with any manual auditing methods.

Auditors are now empowered to possess an automated, simple-to-use method to gain the information they require to meet customer compliancy needs. The easy-to-use interface provides the auditors the toolset they require to audit the important data they need, all from a single, central location. They will have abilities to filter it, based on their unique requirements, and display the data of interest using standard or customized reports.

The IBM Audit Management Expert for z/OS tool provides three different key elements to auditing success:

- ▶ Segregation of duties to ensure complete integrity

- ▶ Centralization of all data to be audited, eliminating the difficult and time consuming collection of data from other systems

- ▶ Cost reduction of auditing and out of compliance risk while pulling together disparate data sources from all the systems through automation

A simple-to-use interface accesses a central repository, giving auditors a comprehensive view of all business activities collected. This is done without the need to rely on technical personnel to perform the monitoring. The product repository can also be audited to provide integrity, which may prevent audit data tampering.

The security of Audit Management Expert is implemented using authorization and collection profiles, to which various privileges can be assigned:

- ▶ Administration of users and passwords are independent of operating systems and DB2 privileged users.
- ▶ Collection profiles provides different access level privileges for users.
- ▶ Collection profiles are used to determine which objects and events can be audited.
- ▶ Authorization profiles are provided to determine which of the collected audit data is visible to end-users (Auditors).

Lastly, DB2 Audit Management Expert for z/OS provides an option to perform log analysis from within the reporting interface. The log analysis function of Audit Management Expert allows you to view who modified audited tables in a DB2 system and to see the actual changes made. An interface is provided to enable the user (auditor) to supply as input, the information needed for log analysis.

We provide detailed information about this tool in Part 4, “DB2 Audit Management Expert” on page 161.

4.2 Data Encryption for IMS and DB2 Databases Tool

DB2 and IMS data encryption constitutes two major functional operations:

- ▶ Encryption
- ▶ Decryption

Conceptually, encryption and decryption is a simple task. Only someone who knows the key can decrypt the data. An encryption key label is assigned by your security administrator. Data is encrypted at the IMS segment and DB2 table level. You can have different encryption exits for different segments or tables. For example, in IMS, a financial application segment could use one exit and a personnel segment another. Log records and image copies of your data are also encrypted.

Data Encryption for IMS and DB2 Databases Tool supports both secure key and clear key encryption and index access is not affected by encryption. Because the EDITPROC is driven for every read and write to the table, unloaded data will be decrypted as it is processed, and re-encrypted as it is loaded.

4.2.1 DB2 encryption

The difference between EDITPROC and Built-in Function for encryption can be summarized as follows:

Data Encryption for IMS and DB2 Databases Tool uses an EDITPROC to encrypt the data row. The benefits of using the EDITPROC methods are as follows:

- ▶ No application changes are required.
- ▶ One key per table or segment specified in the EDITPROC.
- ▶ Encrypted row same length as clear row.
- ▶ Indexes are not encrypted ((because encrypted indexes can impact performance) and the EDITPROC does not support encryption of indexes.
- ▶ Tables with ROWIDs or LOBs cannot be encrypted.

DB2's built-in functions allow the application to encrypt at the column level:

- ▶ The application now determines which columns to encrypt.
- ▶ The application determines which keys to use (on a field basis).
- ▶ Your row size and table layout must change.
- ▶ Indexes are encrypted (This is good for security, but impacts performance negatively).
- ▶ Tables with ROWIDs or LOBs cannot be encrypted.

A DB2 table can only have one EDITPROC exit defined. Should your DB2 table already have an EDITPROC exit specified, and you need to implement Data Encryption for IMS and DB2 Databases Tool, you must code an alternative solution for your existing EDITPROC exit.

Based on the presence of an EDITPROC on the table, DB2 determines that the EDITPROC exit is required. DB2 loads this exit. DB2 calls the exit and passes it the unencrypted row. The exit invokes ICSF services, passing the user-defined data encryption key label (provided by the exit) and the unencrypted row. The key label refers to a DATA, CLRDES, or CLRAES key type that must be predefined by the ICSF administrator.

When the row has been successfully encrypted, the exit passes the row back to DB2. DB2 puts the encrypted row into the table.

We provide detailed information about this tool in Part 5, "Data Encryption for IMS and DB2 Databases Tool" on page 275

4.2.2 IMS encryption

For IMS, the Data Encryption for IMS and DB2 Databases Tool exit routines are implemented at the segment level. The IMS exit routines can be different for each segment. The tool uses the IMS segment edit/compression exit routine.

Implementing data encryption for IMS consists of the following tasks:

- ▶ Define a key.
- ▶ Build an encryption routine that uses the tool to exploit the IMS segment edit/compression exit routine.
- ▶ Unload the database.
- ▶ Reload the database using the encryption exit routine to encrypt the data.

You can implement encryption for IMS with and without compression. You implement data encryption as part of a database unload and reload operation. After you unload a database, and prior to reloading it, include the name of your customized user exit routine in a DBD SEGM statement to encrypt segments that have been specified for encryption.

4.2.3 Data Encryption for IMS and DB2 Databases Tool summary

The IBM Data Encryption for IMS and DB2 Databases Tool is an easily customizable utility. It is also easy to implement on your environment. Few application changes (if any) are required to use the tool, and passwords are not be stored in the application code. Passwords are passed at an exit routine level, and all passwords are managed by the tool.

4.3 Log Analysis Tool

The IBM DB2 Log Analysis Tool for z/OS enables organizations to monitor database changes by automatically building various reports. DB2 logs are processed to uniquely identify who, what, when, where, and how data was changed.

In addition, the tool lets organizations quickly and effectively identify, isolate, and restore unwanted changes while minimizing any downtime. More accurate reporting through functions tools like trigger support and added performance features like the use of a current table space dictionary make the tool more powerful.

To report on data changes, DB2 Log Analysis Tool enables your organization to set filtering criteria for reporting data changes:

- ▶ Database
- ▶ Table space
- ▶ User ID
- ▶ Column data
- ▶ Date ranges and more.

In addition, these filters can be saved to a file for reuse later. The general reports can be loaded back into a DB2 table for retrieving using SQL as well. The auditing feature helps to load detailed data change information into an audit table that tracks when data values hanged and by whom.

The tool integrates with other IBM data management tools. It can be launched from within, and support, the output files of IBM DB2 Object Restore. It can be invoked from the IBM DB2 Administration Tool launchpad. The tool is also functionally used by IBM DB2 Recovery Expert for z/OS.

The Log Analysis Tool requires no other extra security measures outside of standard DB2 security. That is, if a user does not have authority to view a table within a specific DB2 subsystem, Log Analysis Tool will not allow this user to see data changes made to that table. Similarly, undo and redo SQL generated from the product can be run through products like DB2's SPUFI and therefore also adheres to normal DB2 security for the user executing this SQL.

The Log Analysis Tool supports the DB2 Encryption Tool EDITPROC and fully supports data compression. However Data encryption and decryption using ENCRYPT_TDES/DECRYPT_* functions is not supported. DB2's MLS is supported as well. Multi-level security provides authorizations down to the row level within a table. If a user is not allowed to view the row due to MLS definitions, the Log Analysis Tool will not allow the user to view the row either.

4.4 Performance tools

The IBM portfolio of performance solutions provide comprehensive, reliable, and integrated end-to-end monitoring of your DB2 for z/OS environments. Systems and application tuning abilities provided by the tools helps you to monitor, analyze, and optimize the performance of DB2 for z/OS applications in various modes. These include online and immediate real time alerts when problems occur in batch and in reports.

The monitoring tools identify and track problems across your enterprise, allowing you to see information from multiple monitors and third-party software in one single location. This helps you make decisions quickly, efficiently, and proactively.

4.4.1 DB2 Query Monitor

DB2 Query Monitor enables you to identify problem SQL activity and applications. It allows you to focus your efforts on improvement and proactively manage DB2 resources to react quickly and effectively to fix critical DB2 problems.

DB2 Query Monitor allows you to access alerts from a Java-based Web browser. Enhanced integration capabilities let you launch Optimization Service Center.

Note: APAR PK65120 adds integration with Optimization Service Center to the CAE GUI interface of DB2 Query Monitor.

Optimization Center shows the paths DB2 uses to run queries helping you select the most efficient path. DB2 Query Monitor can also launch OM/PE in context and display threads in the active subsystem to save you navigation time.

Support for Audit Management Expert's Audit SQL Collector are provided as well. It allows co-existence of Query Monitor and Audit Management Expert's Audit SQL Collector (ASC) on the same DB2 subsystem.

DB2 Query Monitor requires the use of several DB2 accounting and statistics traces, thereby enabling DB2 Query Monitor to trace and record subsystem data and events for use in problem determination. Accounting traces are records that enable you to trace and record subsystem data and events relating to application programs.

The IBM DB2 auditing solutions help in monitoring the data control, data definition, and data integrity in DB2 environments. Several mechanisms provided by DB2 enable the creation of an audit trail. This trail can be difficult to manage for the DBA and also for the auditors which rely on the DBA for information.

DB2 provides, through the audit trace, a method to record audit data. This feature enables DB2 to use a trace and record audit activities initiated by specific users. When DB2's audit trace is activated, the following type of information can be captured to the trace destination:

- ▶ Authorization failures (GRANTs, REVOKEs, DDL changes and so forth.)
- ▶ Utility executions

DB2 Query Monitor captures or records audit information as well. This information may be written to the output trace destination specified for the audit trace. DB2 trace records can be written to GTF, SMF, or an OP buffer. DB2 auditing requires a trace to be activated. This can quickly become expensive if many tables must be audited. QM does not require these expensive tables. This reduces overhead because it uses the regular processing features of DB2 rather than an additional tracing feature, which increases overhead.

4.4.2 Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS

The Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS (OMPE) tool is a member of the family of the IBM Tivoli OMEGAMON integrated products, which provides an overall monitoring and tuning solution across the several components of the System z platform. From the DB2 subsystem perspective, this is one in a series of complementary solutions that can provide a complete view of the DB2 performance story.

OMPE uses traces to capture DB2 historical data for all trace types. There are several configuration options as part of the setup of the OMEGAMON historical data collector. Several of these options control what types of, and how much, trace data is gathered and stored by the OMEGAMON collector.

Regulatory compliance issues can be costly. It helps if functions are provided by DB2. DB2 9 for z/OS adds a new capability for a trusted context and database roles, so that an application server's shared user ID and password can be limited to work only from a specific physical server. It also allows better auditing and accounting from client systems. Auditing has improved with more granularity in the audit traces that are collected.

About DB2 traces

Each trace record within DB2 has an identifier called an Instrumentation Facility ID (IFCID). Accounting traces track application-level events within the DB2 subsystem. Accounting trace records are written to IFCID 3 and to SMF record ID 101. There are six main classes of accounting traces:

- ▶ Class 1 (Elapsed time)
- ▶ Class 2 (In-DB2 time)
- ▶ Class 3 (Wait times)
- ▶ Class 7 (Package level In-DB2)
- ▶ Class 8 (Package level Wait)
- ▶ Class10 (Package detail trace)

Class 10 can also be used to obtain information about stored procedures debugging. However Class 10 can be expensive and should be only used with care and when needed.

Audit traces are used to monitor and track specific activities on the DB2 subsystem. With audit trace, the more tables that are audited and the more transactions that access them, the larger the overhead. The overhead of audit trace is typically less than 4–5 percent.

The audit trace is useful for installations that must track specific types of DB2 events. Users who want to audit by authid, specific table accesses, and other DB2 events, will find the audit trace providing much value. Some categories of audit information that can be provided are as follows:

- ▶ Instances in which an authorization failure occurs. As an example: A user attempts to SELECT data from a table for which he has not been granted the authority.
- ▶ Executions of DB2 DML - GRANT and REVOKE statements.
- ▶ AUDIT changes or AUDIT ALL DB2 DDL statements on tables.
- ▶ DELETES, INSERTs, and / or UPDATEs for an audited tables.
- ▶ All authid changes resulting from execution of the SET CURRENT SQLID statement.
- ▶ All execution of DB2 utilities.

For additional information about trace overhead, see *DB2 9 for z/OS Performance Topics*, SG24-7473.

Different DB2 traces may be used to perform various levels of analysis. A sound strategy is to use a top-down approach and start with the accounting traces to isolate problem applications. Once the problem has been identified, the next step is to use the performance traces to analyze the application in further depth. Care should be used to run just the trace IFCIDs needed, so as not to incur unnecessary overhead by tracing irrelevant events.

Data security has become critical due to increase of severe data breaches and regulatory compliance requirements. Effective security is multi-layered and challenging. IBM has a rich portfolio of products to address your security and auditing requirements.



Tivoli products

In this chapter, we provide a general introduction on Tivoli security and compliance management products. We describe briefly how these tools help facilitate better security management and meet the compliance and regulatory issues of your organization.

This chapter contains the following sections:

- ▶ “Tivoli zSecure suite” on page 100
- ▶ “Tivoli Security Information and Event Manager” on page 105

5.1 Tivoli zSecure suite

The IBM Tivoli zSecure suite is a set of products that provides a solution to meet your mainframe security administration, audit, and compliance challenges. zSecure suite supports administration and policy enforcement of Resource Access Control Facility (RACF) security and monitoring, auditing and alerting of RACF, ACF2, and top secret-based mainframes.

zSecure suite consists of two groups of products:

- ▶ zSecure Administration
- ▶ zSecure Audit

In this section, we briefly introduce the component products to help you understand the purpose of each tool and how they can improve your security posture, comply with industry regulations, identify and aid in the reduction of audit findings, aid in compliance initiatives, and improve overall efficiency.

5.1.1 zSecure Administration products

The zSecure Administration products are as follows:

- ▶ zSecure Admin
- ▶ zSecure Visual
- ▶ zSecure CICS Toolkit

zSecure Admin

Tivoli zSecure Admin is a next generation security software product that enables efficient and effective IBM RACF administration, user management, and compliance management on the mainframe.

Companies are facing more and more security administration challenges. Many companies do not have experienced RACF administrators to meet their security compliance requirements. It is expensive and time consuming to train employees with low-level skills. By putting a user-friendly layer on top of RACF, zSecure Admin provides a comprehensive, easy to use ISPF interface for low-level RACF administrators. The product generates the required syntax for RACF commands (based on the input from the window). This helps the administrator increase RACF knowledge and skills, while taking the headache out of remembering the various parameters. Generating RACF commands automatically reduces errors that could lead to security exposures or system downtime. zSecure Admin can help automate the recurring work during RACF administration, enabling advanced administrators to focus on higher-value tasks. Figure 5-1 on page 101 shows the main menu of zSecure Admin.

```

Menu  Options  Info  Commands  Setup
-----
                                zSecure Admin+Audit for RACF - Main menu
Option ==> =
More:  +
SE  Setup          Options and input data sets
RA  RACF           RACF Administration
  U  User          User information
  G  Group         Group information
  D  Data set      Data set profiles
  R  Resource      General resource profiles
  S  Settings      Setropts and class settings
  H  Helpdesk     One-panel helpdesk options
  Q  Quick admin   Quick User Administration
  W  Windows       zSecure Visual administration
  1  Access       Access Check
  2  Queued       Display and action on profiles with QUEUED commands
  3  Reports      Reports with profiles and resources
  4  Mass update   Specify mass copy/recreate/delete actions
  5  DIGTCERT     Work with digital certificates
  C  Custom       Custom report
AU  Audit         Audit security and system resources

```

Figure 5-1 zSecure Admin main menu

The product can be used to generate RACF reports and audit security settings by security administrators, security auditors, and who ever has the requirements. This feature helps customers find potential security problems in their systems before they become breaches. For example, as mentioned in 1.3.1, “Payment Card Industry Data Security Standard (PCI DSS)” on page 7, the countermeasure for “Objective A: Build and maintain a secure network”, customers can use zSecure Admin to verify that critical parameters are changed from the well-known vendor supplied values.

zSecure Visual

zSecure Visual, a Microsoft Windows®-based graphical user interface (GUI) for RACF administration, allows RACF administration tasks to be delegated to junior security administrators. zSecure Visual communicates with a server running under z/OS UNIX to perform the native RACF commands. This insulates the zSecure Visual administrator from the complexities of native RACF and TSO/ISPF. Figure 5-2 on page 102 shows the zSecure Visual GUI interface.

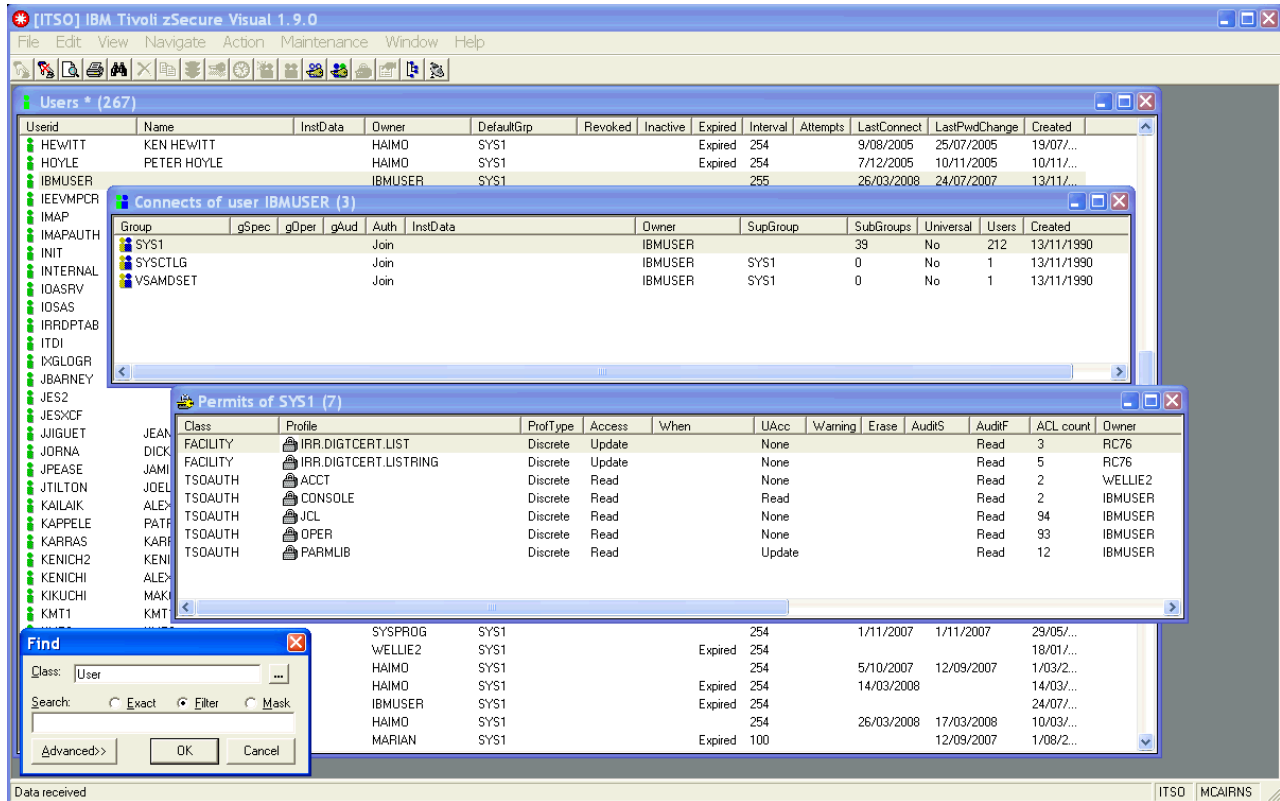


Figure 5-2 zSecure Visual GUI interface

zSecure Visual administrators do not see any RACF commands, so there is no need to remember RACF command syntax. If authorized, the administrator can work with user, group, data set, and general resource profiles without needing to know a single RACF command.

zSecure Visual is an ideal tool for allowing controlled and fully audited RACF administration to take place, without investing large amounts of money in training employees on mainframe and RACF administration. Thus, user administration can be delegated to non-technical users, freeing up precious time within the central security administration team.

zSecure CICS Toolkit

zSecure CICS Toolkit fulfills two major functions:

- ▶ Command interface

The command interface provides the ability to issue RACF commands from CICS. An advantage of using this product is the high performance and efficiency you get from CICS. This is particularly important for organizations which have a large network of security administrators who need to use a common tool to perform RACF administration tasks.

- ▶ Application programming interface (API)

This interface can be used by application designers and programmers to bring external security to CICS and established CICS applications. Thus, access controls and auditing can be managed by RACF rather than relying on embedded security mechanisms within an application.

5.1.2 zSecure Audit Products

The zSecure Audit products are as follows:

- ▶ zSecure Audit
- ▶ zSecure Alert
- ▶ zSecure Command Verifier

zSecure Audit

zSecure Audit is a comprehensive mainframe compliance and audit solution. It enables you to analyze and report on mainframe events, and automatically detect security exposures and mis-configurations. It does this through extensive status auditing and automated analysis using a built-in knowledge base. zSecure Audit has extensive change tracking facilities, which enable you to establish a security baseline and automatically track changes to it. zSecure Audit can correlate information from several different sources and systems, helping you to consolidate reporting and to strengthen controls.

zSecure Audit is generally used by security personnel and internal and external IT auditors to meet compliance and audit requirements. Security personnel use the software for the following purposes:

- ▶ Correlate data from multiple resources, such as RACF database, SMF, z/OS IPL parameters and other configuration information, HTTP logs and flat files, analyze mainframe events, detect security breaches, perform trend analysis, and fix problems.
- ▶ Conduct regular security reviews (control self assessments) to assess the current state of system security.
- ▶ Set up continuous automated auditing to track changes and highlight exposures.
- ▶ Use validation utilities to help maintain a secure and clean RACF database.

zSecure Alert

zSecure Alert is built on zSecure Audit but can run independently. It provides real-time security monitoring capability on mainframes and monitors for intruders and improper configurations. zSecure can take action upon detecting an event, such as revoking a user with excessive violations, or triggering automation to complete a security request. zSecure Alert helps a security team to respond and verify whether the action that triggered the event was appropriate.

Figure 5-3 on page 104 shows the data flow of zSecure. You can see the sources from which zSecure Alert identifies threats and how it generates alerts.

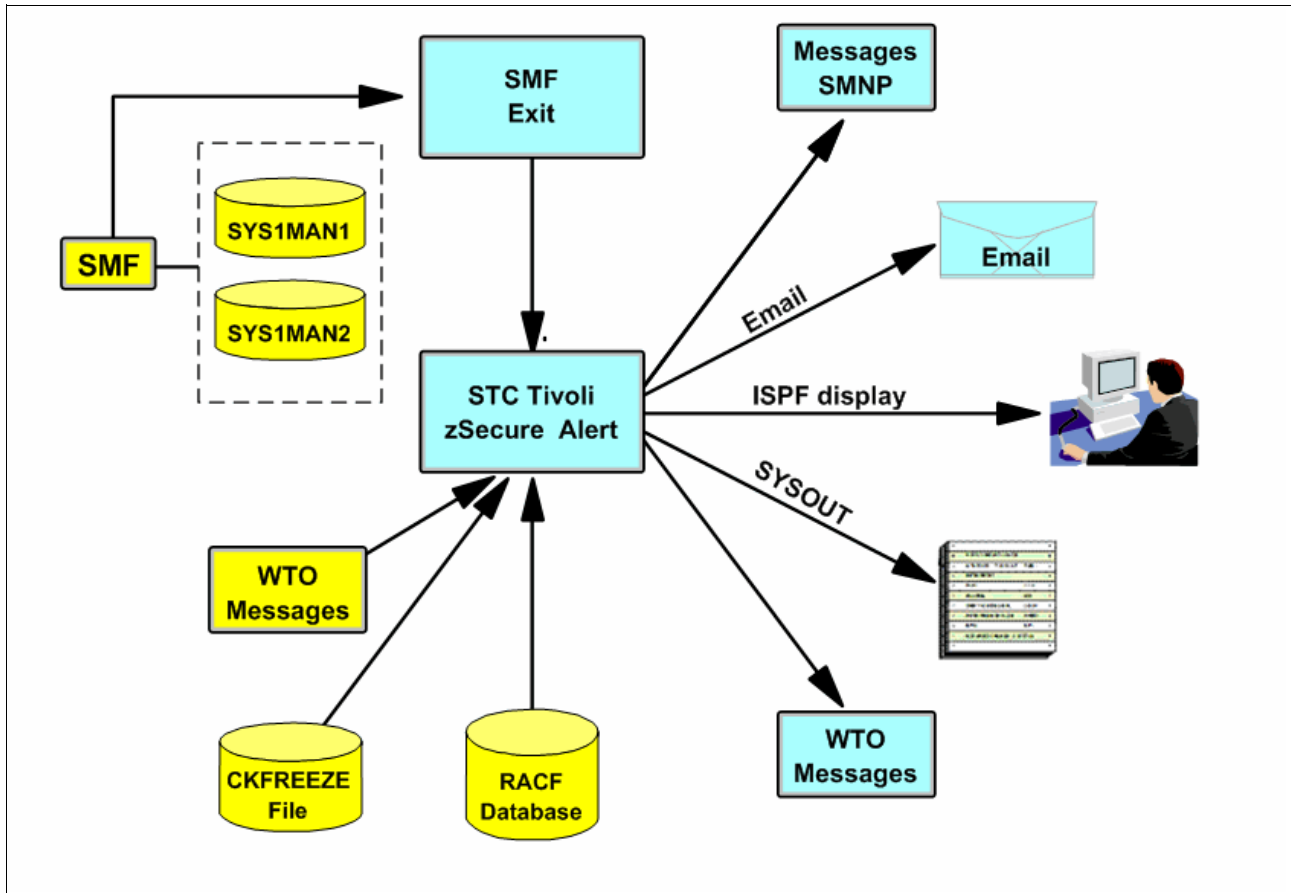


Figure 5-3 zSecure Alert data flow

zSecure Command Verifier

zSecure Command Verifier is an automated policy enforcement solution that adds granular controls for keywords and parameters in RACF commands. It helps enforce mainframe compliance to company and regulatory policies by preventing non-compliant RACF commands, a sample of which is shown in Figure 5-4.

```

READY
CO JPEASE GROUP (SYSPROG)
  You may not connect yourself to group SYSPROG, command terminated
READY
PERMIT 'SYS1.**' GENERIC ID(ALEX01) AC(R)
  Management of locked profiles not allowed, command terminated
READY
ALTDSD 'CKR.**' GENERIC UACC(READ)
  UACC READ setting not allowed, command terminated
READY

```

Figure 5-4 zSecure Command Verifier: Output from non-compliant RACF commands

The use of zSecure Command Verifier enables enforcement of policy dynamically without any coding. Auditors can ask the RACF administrator to define policies using RACF profiles, which can be referenced by zSecure Command Verifier. If a security administrator tries to issue a command that does not comply with policy, the command is either automatically corrected or prevented from being executed.

5.2 Tivoli Security Information and Event Manager

Security information and event management (SIEM) is a primary concern of the CIOs and CISOs in many enterprises. There are two important terms in SIEM:

- ▶ SEM

SEM processes near-real-time data from security devices, network devices, and systems. It targets real-time event management for security operations and effective reporting to internal and external threats.

- ▶ SIM

SIM reports and analyzes data primarily from host systems and applications, and secondarily from security devices. It targets security policy compliance management, internal threat management and regulatory compliance initiatives.

IBM Tivoli Security Information and Event Manager (TSIEM) delivers a comprehensive foundation for addressing your SIEM requirements. TSIEM is comprised of two products:

- ▶ IBM Tivoli Compliance Insight Manager (TCIM)

TCIM handles SIM.

- ▶ IBM Tivoli Security Operations Manager (TSOM)

TSOM handles SEM.

5.2.1 Tivoli Compliance Insight Manager

In this section, we introduce Tivoli Compliance Insight Manager (TCIM), which handles SIM of your enterprise.

TCIM helps the audit and compliance officers gain and maintain an overview of security compliance posture and to monitor user-related security policies. It provides reliable, verifiable log data collection and centralizes security log data from heterogeneous sources. Log data is analyzed and compared with the security policy, and if suspicious activities are detected, TCIM can trigger appropriate actions and alerts automatically.

TCIM has the ability to archive normalized log data for forensic review and to provide consolidated viewing and reporting through a central dashboard. It also provides specific forensic capabilities for searching and retrieving the original log data.

TCIM uses the Generic Event Model (GEM) and the W7 language to consolidate, normalize, and analyze vast amounts of user and system activity. It has the ability to deliver alerts and reports on who touched what information and how those actions might violate external regulations or internal security policies. By revealing who touched what within the organization and comparing that activity to an established policy, security specialists can implement the first layer of defense for information protection, thereby accelerating compliance efforts.

We mentioned policy above, and in 1.3.1, “Payment Card Industry Data Security Standard (PCI DSS)” on page 7, Objective F: Maintain an information security policy, the definition of policy is the key to the compliance management and auditing. TCIM provides template policies in its compliance modules and also the capability for the customer to define policies using its build-in policy definition tools.

TCIM has the auditing capability for z/OS audit data. It uses the event data that is created through normal SMF processing. It copies this data to a file that is stored in z/OS UNIX Services and then passes the data to the Tivoli Compliance Insight Manager. It can capture and process z/OS (including z/OS UNIX), RACF, ACF2, TopSecret, and DB2 SMF data. It can also process the events generated by zSecure Alert.

5.2.2 Tivoli Security Operations Manager

In this section we introduce the Tivoli Security Operations Manager (TSOM).

TSOM is designed to handle SEM. It is targeted for use in the Security Operations Center (SOC) to track and analyze real-time external and internal threats against IT resources. It helps the security officers to collect real-time information of the enterprise, correlate the data within a view to find, and report policy violations or intrusion attempts.

One of the biggest challenges security administrators and analysts face with any enterprise security architecture is finding critical threats and attacks to the infrastructure. Without properly implemented SEM platform, this challenge is extremely difficult due to the following reasons:

- ▶ Disparate point products that are widely distributed.
- ▶ Multi-vendor products without common formats or communications.
- ▶ Inadequate time to manually examine critical logs.
- ▶ No business-relevant context to the data.
- ▶ No inherent link between attack data and host susceptibility.
- ▶ Lack of automation.

TSOM addresses all of these issues. It manages security-related information and logs from a multitude of physical security devices and security software applications, then apply processes required to discern the business relevant incidents in an automatic and efficient manner. TSOM enables security administrators and analysts to analyze and manage security event information in real-time.

TSOM has the ability to gather System z security-related data. It does not have any component residing on System z, it uses conduits to receive z/OS forwarded syslog, SMTP messages, and SNMP traps.

5.2.3 The combined value

TCIM and TSOM can work closely together to help realize the full promise of enterprise SIEM. As a result, IT organizations can lower their exposure to security breaches, collect, analyze, and report on compliance events, and manage the complexity of heterogeneous technologies and infrastructures. This includes support for several hundred applications, host operating systems, security products, network infrastructure, desktops, and mainframe systems.

Figure 5-5 on page 107 shows a typical TSIEM solution for the enterprise, which integrated TCIM and TSOM together. As shown in the figure, TCIM and TSOM can be the data feed to each other. TSOM can be configured to send auditable and correlated events to TCIM. TCIM

reports on these events in compliance and audit reports, and also keeps the event in repository for future reporting, investigation, and so forth. On the other hand, customers can configure TCIM to send alerts of certain events to TSOM. TSOM will raise a ticket to have the alerts recorded and resolved.

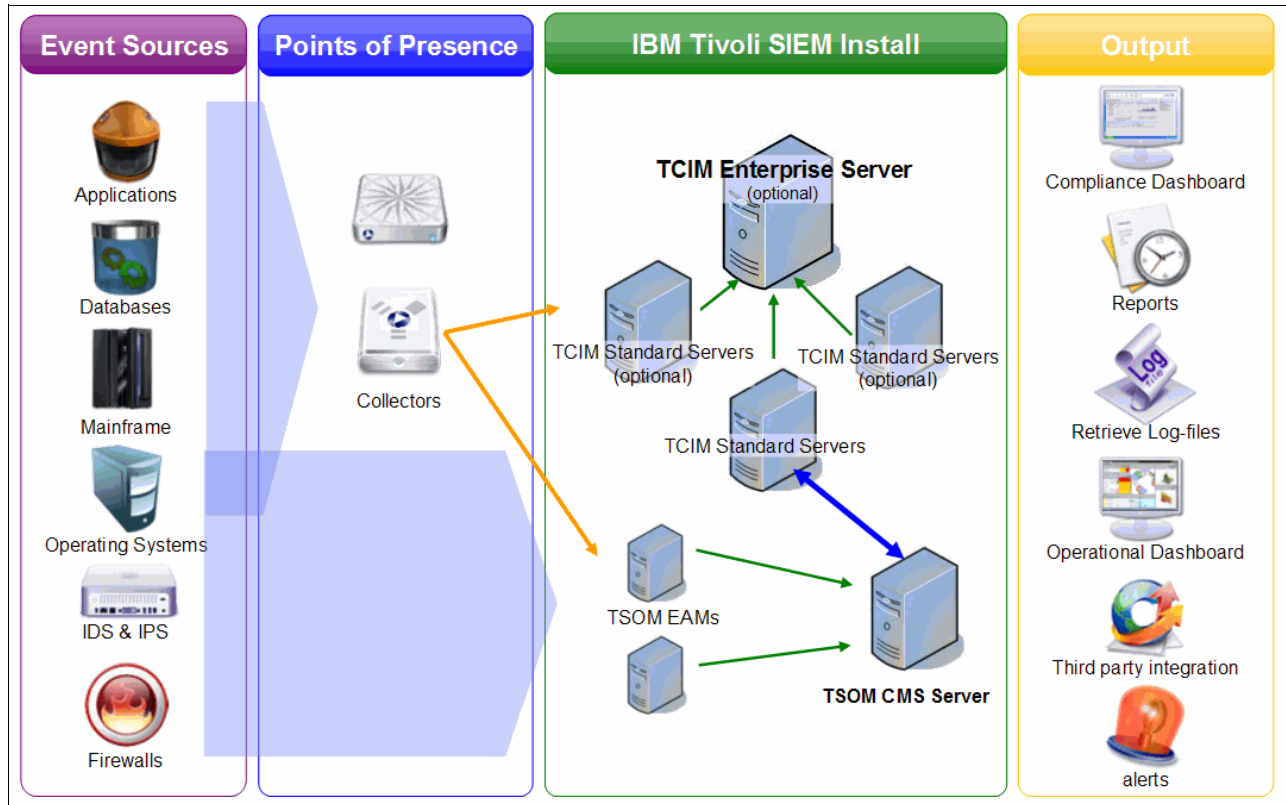


Figure 5-5 TSIEM Solution

Besides the integration of TCIM and TSOM, TSIEM has the ability to work together with Tivoli zSecure Suit products, as mentioned in 5.1, “Tivoli zSecure suite” on page 100. TSIEM can use events generated by Tivoli zSecure Alert to report and alert intruders and improper configurations for RACF and other external security managers for z/OS. TSIEM can also use Tivoli zSecure Audit to monitor and report on privileged users on z/OS operating systems and critical applications.



Optim solutions

IBM Optim™ for z/OS provides facilities that automatically archive, browse, compare, copy, and edit related data. In general, these functions rely upon IBM DB2 relationships, supplemented by user-defined relationships, to manipulate sets of relational data. The following products may be included in your Optim license:

- ▶ IBM Optim Data Growth Solution for z/OS
- ▶ IBM Optim Data Privacy Solution
- ▶ IBM Optim Test Data Management Solution
- ▶ IBM Optim Database Relationship Analyzer

This chapter provides an overview of these components.

6.1 Introduction

IBM Optim enterprise data management solutions focus on critical business issues, such as data growth management, data privacy compliance, test data management, e-discovery, application upgrades, migrations, and retirements. Optim aligns application data management with business objectives to help optimize performance, mitigate risk, and control costs, while delivering capabilities that scale across enterprise applications, databases, and platforms. Today, Optim helps companies across industries worldwide capitalize on the business value of their enterprise applications and databases, with the power to manage enterprise application data through every stage of its life cycle.

Optim Model-Driven Data Governance solutions solve the data management issues, as shown in Figure 6-1.

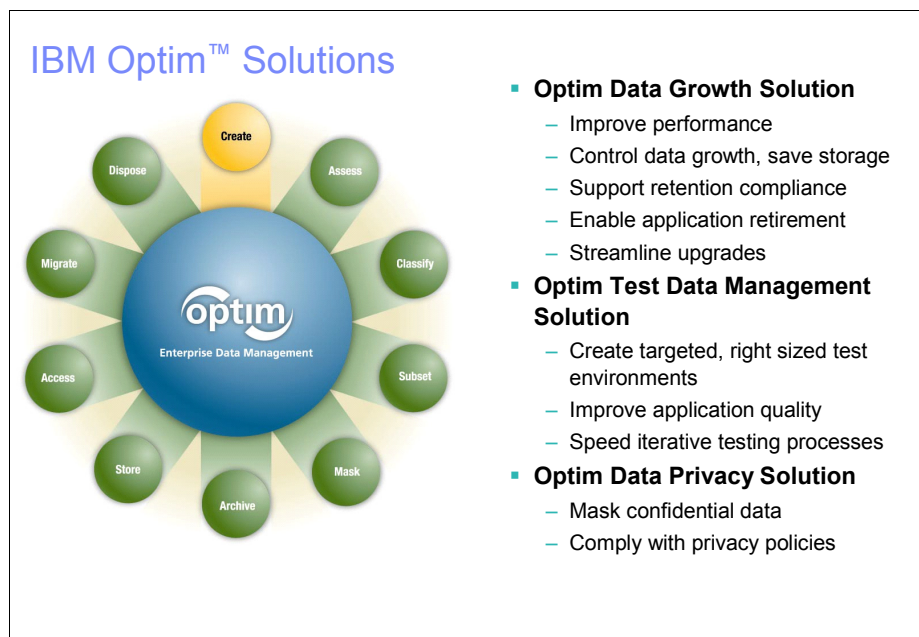


Figure 6-1 The Optim solutions

Optim's broad, heterogeneous enterprise data management solutions help companies solve a variety of business problems, including data growth, test data management, data privacy protection, application upgrades, migrations and retirements, and electronic discovery. Optim provides these capabilities across three major solution areas.

- ▶ The Optim Data Growth Solution mitigates the adverse impact of rapid data growth by archiving historical data safely to a secure archive. The archive can be maintained as a database or as a compressed file, and can be stored on any hardware device. Companies have universal access to the archives, supporting retention compliance initiatives. Archiving seldom-used or historical data streamlines processing workloads, resulting in improved application performance and lower overall storage costs. Archiving prior to an application upgrade reduces the amount of data to be migrated, thereby reducing downtime and enabling timely project completion. Archiving allows companies to retire older or unsupported applications safely while preserving their underlying data records.

- ▶ The Optim Data Privacy Solution offers contextual, application-aware, persistent masking techniques to protect confidential data. By substituting realistic but fictionalized data for private information, Optim creates a protected test database that QA staff can use to generate accurate test results while supporting compliance with privacy mandates such as HIPAA, GLBA, PCI, and the EU Directive on Data Protection.
- ▶ The Optim Test Data Management Solution enables companies to speed the deployment of new or upgraded applications, ultimately accelerating revenue generation. By extracting a precise subset of application data records, companies create targeted, right-sized test databases faster and more easily than by cloning entire production copies. With less data to process, iterative testing cycles are completed more quickly, while testers improve quality by identifying and fixing bugs in the earliest stages of development.

To learn more about IBM Optim enterprise data management solutions, visit the following Web page:

<http://www.ibm.com/software/data/data-management/optim-solutions/>

6.2 IBM Optim Data Growth Solution for z/OS

Enterprise ERP, CRM, and custom applications drive your business initiatives and generate revenue opportunities. Processing more transactions and collecting more customer information is great for business, but un-managed data growth can negatively impact your ability to provide superior service and support. The effects can slow application performance, strain financial and technical resources, and jeopardize completing business-critical processes on time.

IBM Optim offers a data management solution that solves these problems at the source by managing your enterprise application data. Optim provides proven archiving capabilities, allowing users to segregate historical from current data and store it securely and cost-effectively.

By reducing the amount of information in the production database, less disk space is required for application data, which cuts the costs of storage. And because there is less information to sift through, applications process faster. Operations run more efficiently and organizations derive the most business value from mission-critical enterprise applications and their data are the backbone of your business.

With Optim, align application data management with business objectives, optimize performance, control costs, and reduce risks. Optim enables you to segregate historical data from current activity and safely move it to a secure archive. You meet performance targets consistently, which in turn drives revenue growth. Information is available when you need it, so you can respond quickly to client queries and legal requests. Optim enables you to protect the privacy and integrity of your enterprise data, helping you meet compliance initiatives.

Archiving capabilities to achieve SLAs

As a recognized best practice, database archiving segregates inactive application data from current activity and safely moves it to a secure archive. Streamlined databases reclaim capacity and help improve application performance and availability. See Figure 6-2 on page 112.

With Optim, you can establish distinct service levels for each class of application data (for example, current data, reporting data and historical data) and consistently achieve performance targets. Policy-driven archive processes allow you to specify the business rules and criteria for archiving. Criteria are commonly based on functional requirements or data

retention regulations, such as age, date, transaction status, business unit, or company. For example, you may choose to archive all closed orders that are two years old or more. Optim identifies all transactions that meet these criteria and moves them into an archive.

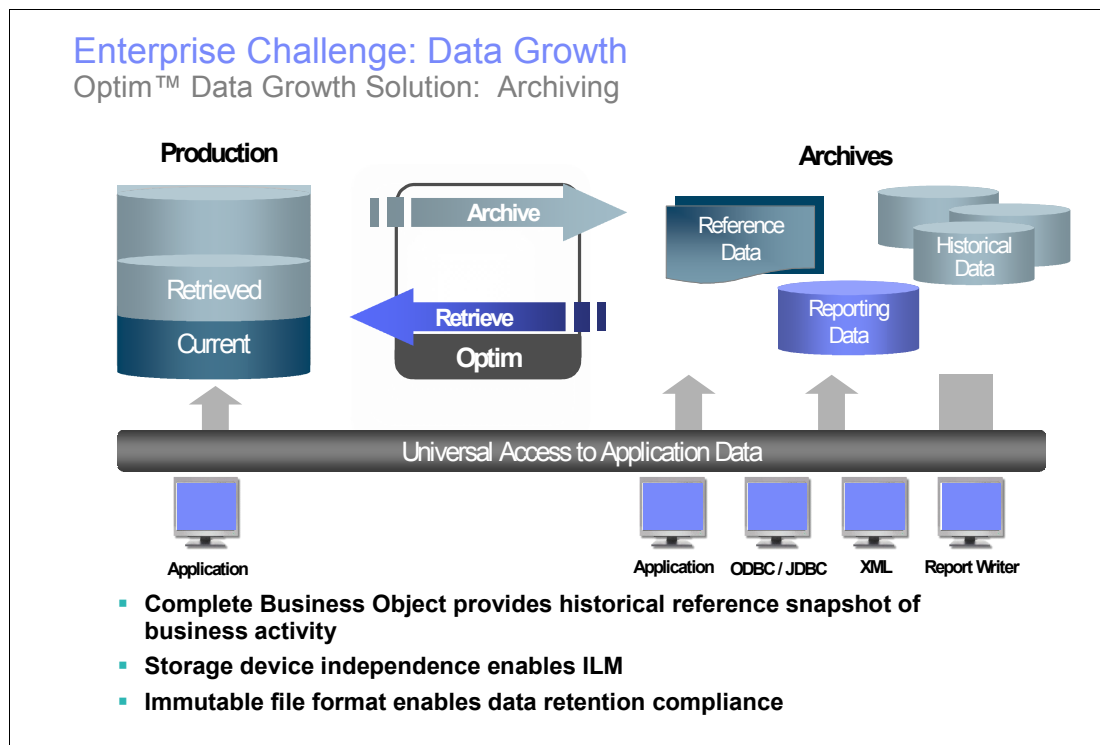


Figure 6-2 Optim data growth archiving

Optim's archive processing capabilities manage application data at the transaction or business object level (for example, vouchers or journals). Archiving preserves the data integrity and includes essential metadata, so that each archived business object represents a historical reference snapshot of business activity. Ongoing, scheduled archiving allows you to tier transactions between the production database and archives, so it is easier to meet service level goals in every case

Data on demand

If you need access to your historical business data to make decisions, run reports, and respond to customer inquiries, audit, or e-discovery requests, Optim lets you choose the most effective access method, based on convenience and cost. You can implement tiered storage strategies to manage application data based on its evolving business value and access requirements. Current transactions remain in the high-performance OLTP environment. Reporting data in history tables can be maintained in mid-tier storage, so you control costs while still meeting service requirements.

To further reduce costs, you can store historical or reference data offline to tape or other long-term storage devices. Maintaining reference data in an immutable format on a secure WORM (Write Once, Read Many) device enables you to protect archived business objects for regulatory compliance. If an auditor should come to call, you will be prepared with complete snapshots of your transactions, perfectly preserved at each point in time. Keep archived business transactions accessible until legal retention periods expire and archives can be deleted.

Reporting on historical information takes less time and effort with Optim's universal access methods; from application-based access (through your current interface) to application independent access, leveraging industry standard methods and tools. Application-based access offers a consolidated view of current and historical information through the existing application interface. Optim also provides application-independent access to archived transactions, without impairing online transaction processing (OLTP) performance, using industry standards methods, such as ODBC/JDBC, XML, or SQL and reporting tools, such as IBM Cognos®, Crystal Reports, Oracle Discoverer, or Business Objects.

Governance and recovery initiatives

Protecting your company from legal and other liability is critical. Optim's data management capabilities allow you apply business policies to govern data retention and disposal. Applying suitable and secure methods for data disposal allows you to prevent your information assets from becoming liabilities. You can automate data retention to support compliance initiatives and respond quickly and accurately to audit and discovery requests. And in the event of a disaster, employing a staged recovery strategy helps ensure the continuity of your business.

Reduce risk and control costs

Optim provides capabilities for controlling your application data from creation to disposal. You can simplify enterprise data management to accelerate business-critical projects. Provide universal access to current and archived data, complete easier upgrades, implement cost-effective tiered storage strategies, and profit from superior application performance and availability. By implementing a proven enterprise data management strategy, you take command of your mission-critical data throughout its entire life cycle and realize measurable benefits across your organization.

6.3 IBM Optim Data Privacy Solution

Safeguarding the privacy of client data is not just good business, it is required to do business. De-identifying confidential data is one of the best ways to protect privacy and support compliance with regulations like HIPAA, DDP, PIPEDA, PCI DSS, and others. Optim delivers powerful data transformation capabilities to mask confidential corporate data, so that you can use it safely for application testing. Safeguard vulnerable test environments by applying simple data masking techniques or pre-packaged transformation algorithms for complex data elements like credit card numbers, e-mail addresses, and national identifiers.

IBM Optim Data Privacy Solution aligns application data management with your business objectives to optimize performance, control costs and reduce risk

- ▶ Protect the privacy of confidential data across non-production environments
- ▶ Apply predefined masking techniques to speed time to delivery
- ▶ Preserve the integrity of the data, while protecting privacy
- ▶ Improve flexibility for masking data in existing non-production databases
- ▶ Support privacy regulations and corporate governance standards
- ▶ Provide a single, scalable enterprise data management solution across applications, databases (DB2, IMS, VSAM/SEQ, Adabas, IDS, Oracle, Sybase, SQL Server®, XML), and platforms (Windows, Unix/Linux, z/OS)

Data privacy compliance

Companies worldwide are subject to government regulations enacted to protect confidential information from misuse. For example, the European Union has established the Personal Data Protection Directive as the framework for privacy protection governing its member countries. In Canada, organizations follow the provisions of the Personal Information

Protection and Electronic Documents Act (PIPEDA), while Australian companies are subject to the Privacy Amendment Act. In the US, a number of regulations apply at the national and state levels. Similar statutes exist worldwide.

Additionally, industry coalitions are developing sector-specific governance standards. For instance, the Payment Card Industry Data Security Standard (PCI DSS), initiated by Visa and MasterCard, is being adopted by other payment card companies in response to the overwhelming incidence of data theft and fraud. The standard requires members, merchants, and service providers to apply 12 security safeguards for the protection of cardholder data. In particular, PCI requirement 6.3.4 states that test databases must not contain personal account numbers (PANs) from production data.

More companies are gaining an understanding of the vulnerabilities across application environments. The same methods that protect data in production, such as access controls and authentication schemes, and network, application and database-level security, may not meet the unique requirements for protecting non-production (development, testing, and training) environments.

Companies spend a great deal of time and money to secure their systems from external attacks, but many do not realize that 70% of data breaches are from internal sources. Examples range from employees, who misuse payment card numbers and other sensitive information, to those who save confidential data on laptops that are stolen or misappropriated. Furthermore, outsourcing application development and testing activities makes it difficult to control access to sensitive data.

Privacy in non-production environments

In most cases, realistic data is required to test application functionality and to ensure accuracy and reliability. Testing environments are often created by simply cloning copies of the production database. This means that sensitive information is propagated from a secure production environment to vulnerable non-production environments. However, although using realistic data is essential for quality application testing, capabilities for *de-identifying* or masking production data offer a best practice approach for protecting privacy.

De-identifying data is the process of systematically removing, masking, or transforming data elements that could be used to identify an individual. Data de-identification enables developers, testers, and trainers to use realistic data and produce valid results, while still complying with privacy protection rules. Data that has been scrubbed or cleansed in such a manner is generally considered acceptable to use in non-production environments. Once the data is masked, even if it is stolen, or lost, it will not be an exposure to privacy.

Data masking can be complex. From a technical perspective, data masking is not a one-time process and must be appropriate for existing development, testing and training requirements. From a business prospective, a single-point solution is not the most cost-effective approach. The ideal solution must support data privacy compliance across applications, databases, operating systems and hardware platforms.

Protect privacy

The IBM Optim Data Privacy Solution provides comprehensive capabilities for de-identifying application data that can be used effectively across non-production environments. You can take the necessary steps to protect privacy, and still provide the necessary realistic data for use in development, testing, training, or other legitimate business purposes.

Optim's scalable data masking techniques can be deployed across applications, databases, operating systems, and hardware platforms to meet your current and future needs. The masked test data still makes sense to developers, testers and trainers and produces accurate, reliable results, but is worthless to thieves and hackers.

Implement data masking techniques

Using Optim, developers and testers can apply a variety of proven data transformation techniques to substitute confidential data with contextually accurate, but fictionalized data to produce valid results. With support for the leading database management systems, Optim also provides federated access capabilities that allow you to extract and mask appropriate data from multiple production data sources in a single process.

In addition, Optim's *mask-in-place* capability allows you to mask data that was extracted using third-party tools or data that already resides in other non-production environments. Organizations that have data in place for testing, or that use backup type facilities to create those test databases can benefit from Optim's mask-in-place capabilities. Using Optim to mask data directly at the source eliminates the need to move the data for additional processing and still preserves the referential integrity of the data.

Optim's application-aware masking capabilities help ensure that masked data, like names and street addresses, resembles the look and feel of the original information. Optim preserves the integrity of the data and produces consistent and valid results that reflect the application logic. For example, surnames can be replaced with random surnames, not with meaningless text strings.

Context-aware, prepackaged data masking routines make it easy to de-identify many types of sensitive information, such as birth dates, bank account numbers, national identifiers (like Canada's Social Insurance numbers or Italy's Codice Fiscale), benefits information, health insurance identification numbers, and so on. Some examples of Optim's masking techniques include substrings, arithmetic expressions, random or sequential number generation, date aging, and concatenation.

Optim's Transformation Library routines allow for accurately masking complex data elements, such as Social Security numbers, credit card numbers, and e-mail addresses. Built-in lookup tables support masking names and addresses. You can also incorporate site-specific data transformation routines that integrate processing logic from multiple related applications and databases and provide greater flexibility and creativity in supporting even the most complex data masking requirements.

Each of the methods described so far is effective for masking data to safeguard confidentiality. However, with relational database applications you need the capability to propagate a masked data element to all related tables in the database to maintain the referential integrity. For example, if a masked data element, such as a telephone number, is a primary or foreign key in a database table relationship, this newly masked data value must be propagated to all related tables in the database or across data sources.

Key propagation helps preserve the referential integrity of the transformed data across applications, databases, and operating environments. Without key propagation, the relationships between parent and child tables would be severed, causing the test data to be inaccurate. Consequently, application testing will produce unreliable results. Optim's persistent masking capabilities propagate masked replacement values consistently and accurately across multiple data sources to generate valid test results.

See Figure 6-3 for Optim Test Data Management's set of data masking techniques.

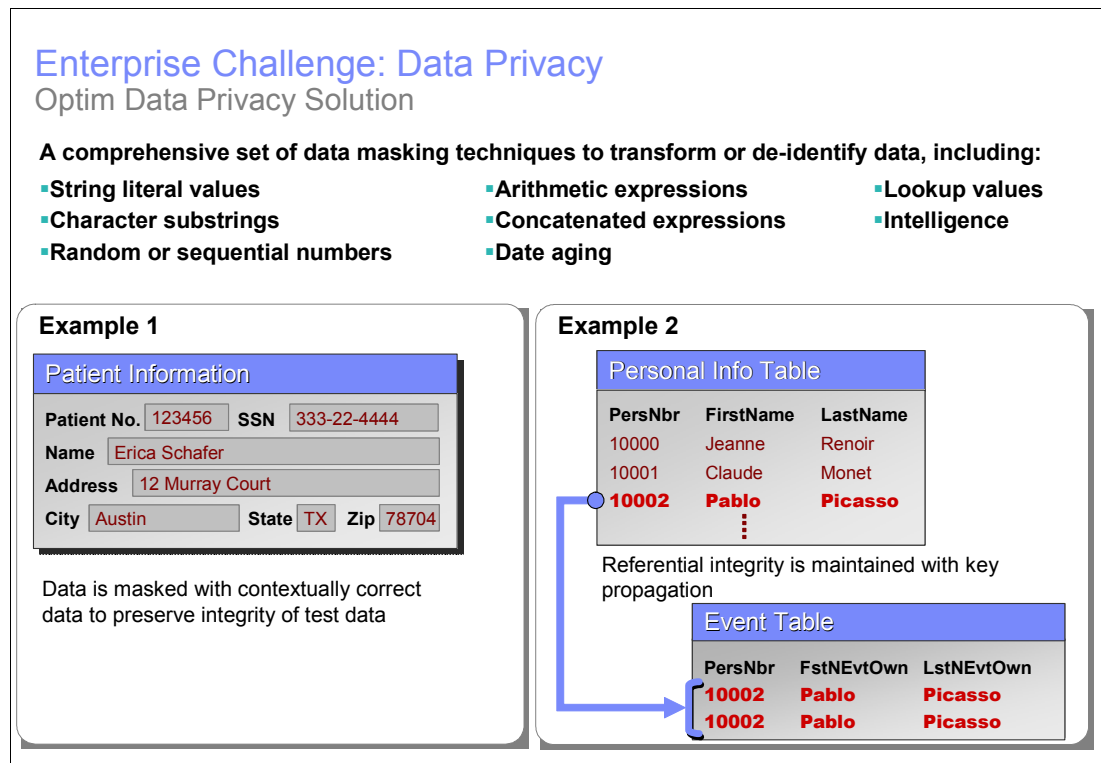


Figure 6-3 Optim Data Privacy data masking techniques

Support your data privacy across environments

Implementing Optim helps you comply with data privacy regulations and protect the confidentiality of sensitive information across your enterprise. You benefit from Optim support of all leading enterprise databases and operating systems, including IBM DB2, Oracle, Sybase, Microsoft SQL Server, IBM Informix, IMS., VSAM, Teradata, Adabas, Microsoft Windows, UNIX, Linux, and z/OS. In addition, Optim also supports the key ERP and CRM applications in use today: SAP Applications, Oracle E.Business Suite, PeopleSoft Enterprise, JD Edwards® EnterpriseOne, Siebel, and Amdocs CRM.

6.4 IBM Optim Test Data Management Solution

Your organization depends on business-critical applications to drive results. But it is a challenge to stay within tight budgets, while striving to speed the deployment of new applications, upgrades, and enhancements. Creating realistic and consistent application development and testing environments is the first step in delivering reliable applications, enhancements, and upgrades. However, cloning large production databases for testing purposes increases the time needed to run test cases. In addition, cloned data may not support the specific error and boundary conditions required for effective testing. Special test cases may be required before testing can begin. Lastly, manually validating test results is often time consuming and error prone.

Your goal is to deliver reliable application functionality to support operational best practices, stay competitive, and generate revenue.

The IBM Optim Test Data Management Solution offers proven technology to optimize and automate processes that create and manage data in non-production (testing, development, and training) environments. Developers and quality assurance testers can create realistic, right-sized test databases, create targeted test scenarios, protect privacy, and compare before and after test results with speed and accuracy.

Optim’s capabilities for creating and managing test data enable sites to save valuable processing time, ensure consistency, and reduce costs throughout the application lifecycle. With Test Data Management Solution, you can perform the following functions:

- ▶ Apply subsetting capabilities to create referentially intact, right-sized test databases
- ▶ Create targeted test scenarios to force error and boundary conditions
- ▶ Automate test result comparisons to identify hidden errors
- ▶ Easily refresh and maintain production-like test environments
- ▶ Shorten iterative testing cycles and accelerate time to market

As shown in Figure 6-4, we begin with a production system or clone of production. Optim extracts the desired data records, based on user specifications, and safely copies them to a compressed file, IT loads the file into the target Development, Test, or QA environment.

After running tests, IT can compare the results against the baseline data to validate results and identify any errors. They can refresh the database simply by re-inserting the extract file, thereby ensuring consistency.

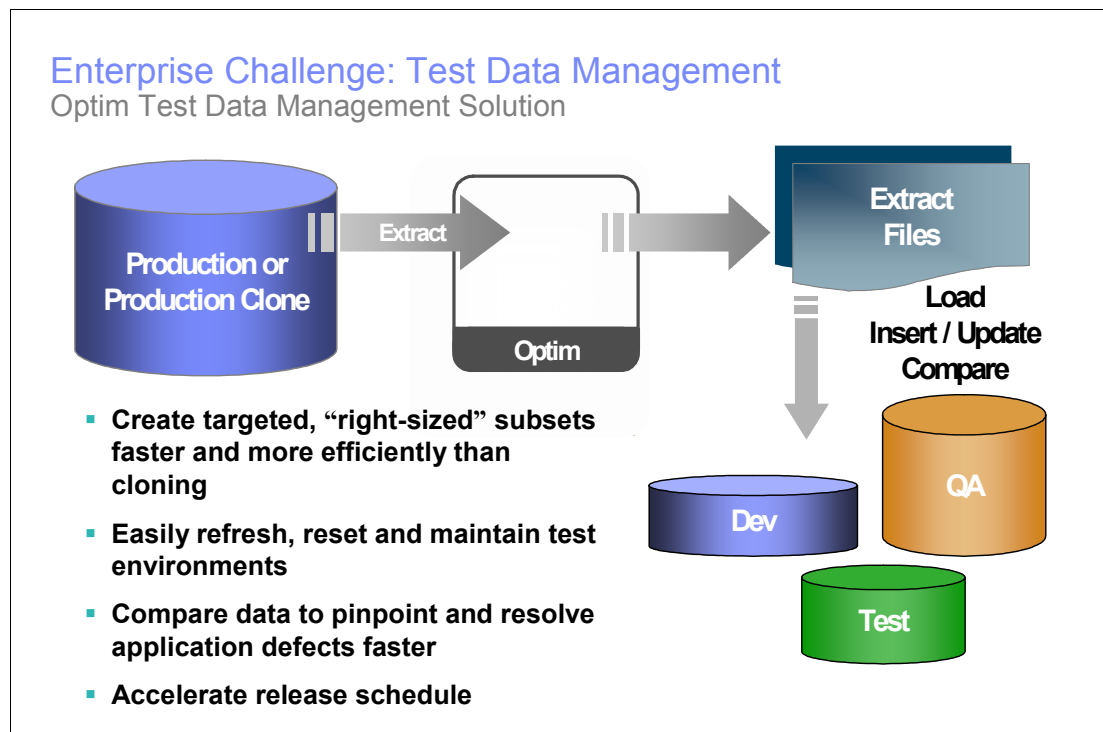


Figure 6-4 IBM Optim Test Data Management in action

Create environments for application testing

In contrast to cloning large production databases, a more effective alternative is to implement test data management and subsetting capabilities that minimize storage requirements, while expanding test coverage. It is much faster to test with smaller, realistic subsets that accurately reflect the production data, without adding overhead to the testing process.

Optim provides the capabilities to create referentially intact subsets of application data needed for testing. Optim Database Relationship Analyzer (see 6.5, “IBM Optim Database Relationship Analyzer” on page 120) represents a unique, proven technology that identifies, navigates, and intelligently processes data relationships and extracts precise subsets of related data, no matter how complex the data model. Subsetting capabilities allow you to accurately capture realistic application test data and reduce capacity requirements for multiple testing environments. Optim also provides capabilities for capturing related data among multiple integrated applications and databases. This federated extract capability makes it easy to create production-like environments that accurately reflect your end-to-end business processes.

Using Optim to define subsetting selection criteria is as easy as selecting the tables, relationships, and other functional criteria needed to cover a specific test case. Optim supports data relationships defined in the database, and those enforced through the application logic. Once subsetting selection criteria are defined, Optim’s processing capabilities identify and extract the precise subsets of data. Insert and load options allow you to populate or refresh test databases efficiently and accurately, with no impact on production. You can also save the processing specifications and results in files that can be shared, easily modified, and reused.

For example, it is easy to populate a test environment with referentially intact customer and order information for a specific business unit, fiscal year, or equivalent criteria chosen by the business user. To refresh the test environment, you can insert or load baseline data. Alternatively, you can extract new records from production to expand test coverage or obtain data for a unique test case. You can create consistent development, testing, or training environments by extracting a standard set of sample data and resetting the baseline after each session.

Test data management and subsetting capabilities help control the size of development and testing environments. Eliminating excess data volume reduces storage requirements and trims costs. You can create any number of right-sized development, test, and training databases to satisfy specific requirements, improving both coverage and accuracy. Streamlined test databases are easier to manage and maintain, so you can speed iterative testing cycles and shorten the time necessary to deploy new application functionality.

Protect privacy in test environments

Optim also offers a variety of methods for masking test data to protect privacy and support regulatory compliance initiatives. Unique table and column mapping capabilities help you map specific target test data. Sites can use context-aware data masking routines to de-identify key data elements across your applications. Optim captures and accurately processes application data elements so that the masked data does not violate application logic and produces valid results.

You can combine test data management capabilities with comprehensive data masking capabilities available in the IBM Optim Data Privacy Solution (see 6.3, “IBM Optim Data Privacy Solution” on page 113). This solution includes built-in lookup tables and prepackaged routines that support transforming complex data elements, such as Social Security numbers, credit card numbers, and e-mail addresses. You can also incorporate site-specific data transformation routines, integrating the processing logic from multiple related applications and databases.

Force error conditions

Another way to optimize your testing environment is to create targeted test scenarios. Optim includes comprehensive relational editing capabilities that not only simplify the tasks necessary to create this special data, but also make it easy to browse data and resolve

application errors. A simplistic, one-dimensional view of your test data is insufficient. Optim provides capabilities for browsing and editing data in its relational or business context across multiple tables, which offers a better way to envision the data relationships and structure of the application data model.

Practical and intuitive commands simplify relational editing and ensure data integrity. You can easily edit test data to trigger error or boundary conditions and to verify exception handling during testing. For example, if diagnostic codes are four digits, and range in value from 0001 to 1000, then a forced value of 2000 would trigger an error in the context of the application test. A powerful undo capability allows you to reverse an unlimited number of editing changes. A sophisticated audit facility tracks changes and saves details for review by authorized users.

Automate data comparisons and analyze results

The ability to analyze and validate test results is critical for ensuring application quality. The database size and complexity significantly increase the effort involved in examining test results. See Figure 6-5 for the summary of functions seen so far.

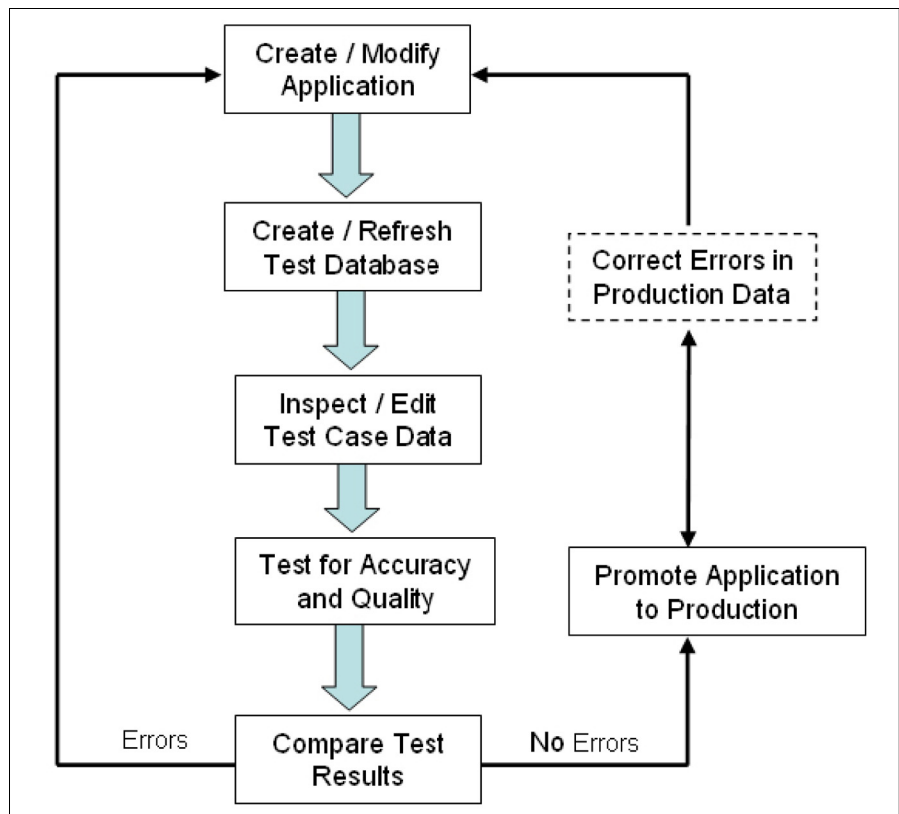


Figure 6-5 IBM Optim improves every stage of the application testing process

After a test run, Optim analyzes the before and after images of the test data, automatically detects any differences, and presents the results in a concise report. You can browse comparison results in a highlighted display for easy analysis, thereby saving countless hours.

An intuitive, online interface and full-function browse utility eliminate time-consuming and error-prone table-by-table comparisons. Optim not only identifies the expected database changes, but also uncovers differences that might otherwise go undetected. Those application defects that are hidden or difficult to trace can be identified quickly and resolved at a fraction of the cost. Compare specifications can be saved, reused, and shared to improve consistency and productivity.

You define your business objectives and processes related to enterprise application data, and then quickly apply Optim processes and technology to improve all phases of application testing.

Optim provides a central data management solution that scales to meet enterprise needs. In addition to supporting your custom and packaged applications, Optim is the only solution to provide a consistent test data management approach across the leading ERP and CRM applications: Oracle E-Business Suite, PeopleSoft Enterprise, JD Edwards EnterpriseOne, Siebel CRM and Amdocs CRM. And it supports all major enterprise databases and operating systems: DB2, Oracle, Sybase, Microsoft SQL Server, Informix, IMS, VSAM, Microsoft Windows, UNIX, Linux, and z/OS.

6.5 IBM Optim Database Relationship Analyzer

Most of the ERP, CRM, and custom applications that drive your business operations rely on complex relational database technology. It is not uncommon for application databases to contain dozens, hundreds, or even thousands of database tables and just as many data relationships. Database administrators are often challenged to navigate through the database schema to ensure the referential integrity of the data each time a new application enhancement or upgrade is developed. Figure 6-6 summarizes the challenge in capturing consistent data for testing or archiving purposes.

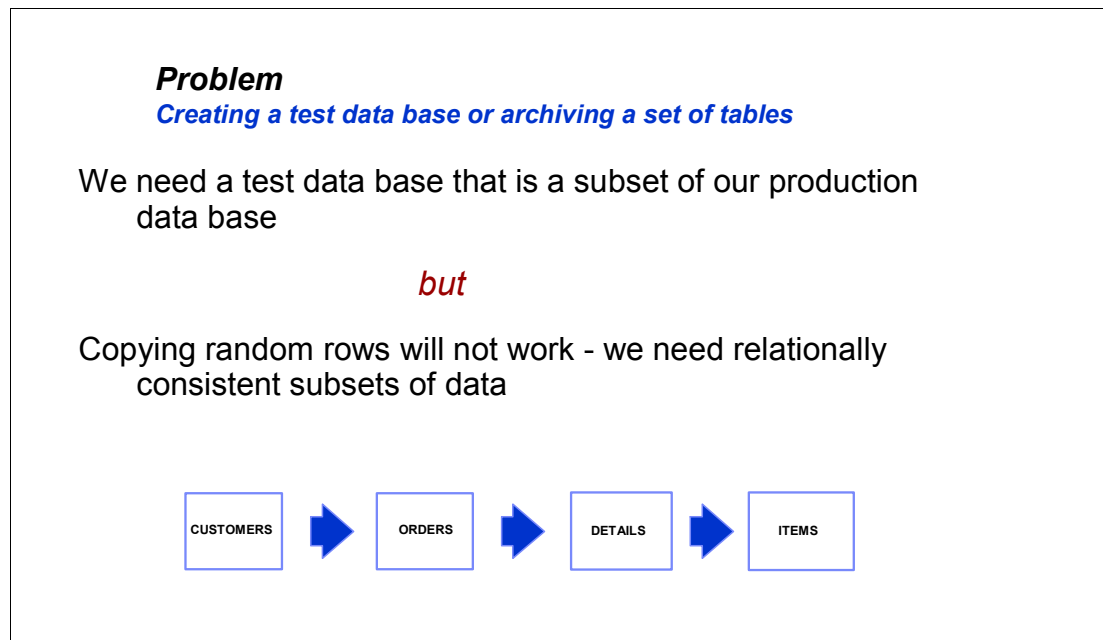


Figure 6-6 Need for a way to capture relationally consistent data

The complexity of relational databases is not limited to large-scale systems, even a simple database with a few tables can support complex relationships and be accessed from other applications, making it more challenging for DBA's to navigate and manage database relationships effectively.

Normalized relational data, by its nature, is fragmented, making it difficult to recognize related information, let alone manipulate it. The referential integrity of the data must remain intact, but DBAs cannot be aware of every possible database relationship. Complete intelligence about data relationships is rarely fully defined in the database management system (DBMS) catalog

through explicit referential integrity. Data relationships often exist outside the DBMS. Real-world data relationships are defined and enforced by the application logic and yet must be understood and maintained. Adding to the complexity is the challenge of managing data relationships across heterogeneous database management systems, legacy data, and differing database schemas.

Without access to a generalized technology, there is no easy way to determine how an anticipated change to the application database schema will impact the referential integrity of the data. Related data must be analyzed and understood to make its business use in the enterprise application more apparent. Once these challenges are addressed, accuracy is improved, data is more usable, and it is easier to implement enterprise data management strategies, such as database archiving, test data management, data masking, and more.

The IBM Optim Database Relationship Analyzer provides the capabilities to discover, record, refine, and manage groups of database tables and relationships that support a single business application or an entire enterprise application environment. This Optim function is integrated in the Optim Solutions and it will help you perform the following functions:

- ▶ Automatically analyze database relationships across applications
- ▶ Discover all, or specific database relationships, based on your parameters
- ▶ Identify hard-to-find relationships defined and enforced by the application logic
- ▶ Compare database images to validate changes and preserve data integrity
- ▶ Promote consistent database administrative activities

Optim provides a complete view of database relationships, essential for planning application upgrades, initiating database and data relationship changes, enhancing or adding new application functionality, and database cleanup, migration, and testing. See Figure 6-7.

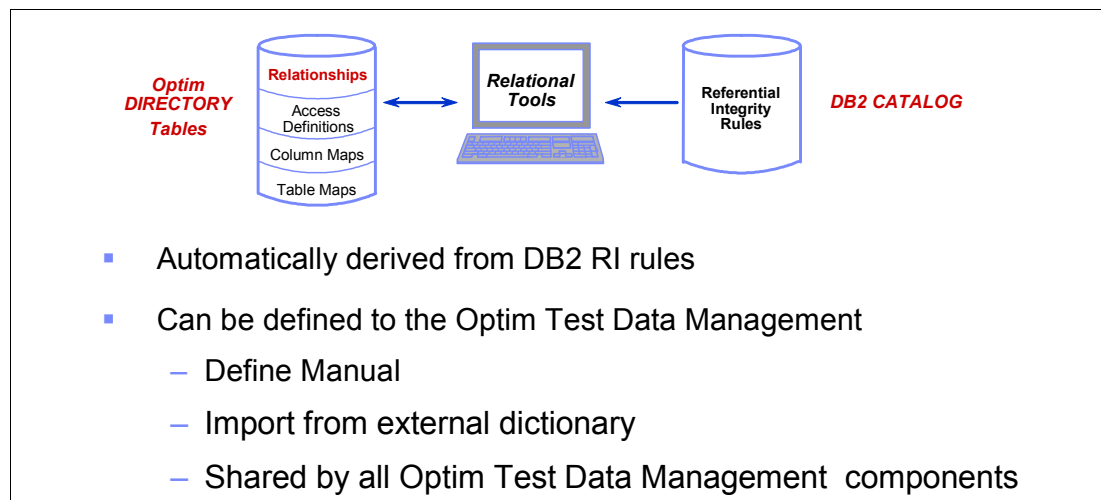


Figure 6-7 Dealing with relationships

The Optim Data Relationship Analyzer discovers database relationships based on your search parameters. You can discover and save the database relationship information as a *group* and manage these relationship groups across your enterprise application environment. For example, discover relationships that are hard to find and manage, such as those associated with monthly or year-end applications and dynamic or transient applications. Identify relationship groupings to preserve the referential integrity of the data for other data management activities, such as archiving, test data management, data masking, backup and recovery, and replication with other tools. Group related tables based on how they are used in the application environment (For example inventory, sales and payroll, and other uses).

The Optim Database Relationship Analyzer includes the following components:

- ▶ Group Discovery
- ▶ Trace Analyzer
- ▶ Integration APIs

Group Discovery process

The Group Discovery process enables finding the database relationships based on a user-defined set of parameters. For example, you can specify a starting point table, boundary objects, tables, and relationships to ignore, and additional relationships to find. The discovery process can be run to analyze the entire database catalog or discover the relationships specific to a selected starting point table. You can also take advantage of the group validation feature to compare versions of a table/relationship group to determine if any changes have occurred in the DBMS environment or in the applications that reference the tables in the groups.

For example, you need to add a salary table to a payroll application and you want to see the impact that change will have on the other tables. You can use the group discovery process to obtain a before and after view of the database and then perform a group validation to automatically pinpoint the differences. First, run the group discovery process on the original database to get a baseline. Then add the salary table to the database and run the group discovery process again to get a current version and view of related groups. Next, select that new version to compare to the baseline. The differences display automatically.

Trace Analyzer process

The Trace Analyzer complements the Group Discovery process to find database relationships that are enforced by the application and not defined in the database catalog. This function assesses the SQL trace information and parses the SQL (DDL and DML) statements that were issued during a specific application run period. Optim considers tables to be related if they are referenced in the SQL statements that were issued by the same application. The Trace Analyzer supports DDL and DML statements, such as CREATE, ALTER, INSERT, DELETE, UPDATE, and SELECT.

For example, you make changes to the data schema and want determine if any external SQL applications, such as the payroll application, access the tables that you have deleted or renamed, so you can update the source code. First, run the SQL trace on your database for a period of time when it would be accessed by the external applications. Next, run the Trace Analyzer based on the SQL trace information. Finally, run the Group Discovery process using input from the Trace Analyzer to discover the relationships between the updated payroll tables and the external applications.

Integration APIs

Integration API contains a set of SQL views and stored procedures that are defined on the Optim Database Relationship Analyzer metadata database and enable retrieving group and table relationship information. You can use these views and stored procedures to retrieve group information and table relationship information. In fact, any DBMS tool or application can use the views to retrieve information that might be helpful in managing your environment. DBMS applications can call these stored procedures. Optim provides sample code statements that you can use to create a sample program to call integrated stored procedures.

View and account for tables relationships

The capability to discover and understand the data relationships within a single application or across integrated applications offers several benefits. The capability to find table relationships can help you perform impact analysis across databases before deleting or modifying database elements (for example, renaming table columns).

The group validation feature helps analyze relationship information before and after changes are made (such as deleting or adding a table and validating runtime table interactions). The SQL trace analysis feature helps you find application enforced data relationships that are not defined in the DBMS catalog and makes it easier to observe the runtime table interactions of external applications that access your database. Understanding the data relationships makes it easier to identify and remove duplicate or unused objects to optimize the database and help improve performance. A clear picture of the data relationships can also help preserve the referential integrity.

The relationship data is used to fast track various data management practices including archiving, backup and recovery, creating practical testing environments, and understanding relationships in replication subscriptions.

Through integration with the IBM Optim Data Growth Solution, IBM Optim Test Data Management Solution, and the IBM Optim Data Privacy Solution, the Optim Database Relationship Analyzer provides the capabilities to analyze interrelated application database environments in less time, migrating referentially intact and complete sets of related data to reduce risk, and improving database management to lower costs.



Part 3

System z synergy

Any application that makes use of cryptography to protect data invokes the System z functions that support the cryptographic key.

The Integrated Cryptographic Service Facility (ICSF) is the software element of z/OS that manages a key securely stored on a cryptographic coprocessor. The instructions for encrypting and decrypting are supported by the hardware platform.

In this part we introduce the System z and z/OS-provided functions and show how they establish the foundation for protecting DB2 for z/OS data.

This part includes the following chapters:

- ▶ Chapter 7, “System z security features” on page 127
- ▶ Chapter 8, “z/OS security” on page 151



System z security features

As we all know, security is one of the hottest topics in the industry these days. You may have heard in the news about data cartridges being lost in transit and legislation demanding higher levels of accountability. Because of unfortunate events like these, security has become a front and center topic in every IT shop.

Not surprisingly, there is no better platform to handle the challenges of keeping your data secure than System z. System z has a history of ensuring that any and all data residing on it is safe and secure. This goes far beyond the usual measures of other platforms.

System z takes a holistic approach to security and is designed so that every component works flawlessly. Each component cooperates with the system as a whole to provide safety for your most important revenue generating items.

Everything from the data coming through your network to the people allowed to access that data is closely monitored, thus providing end-to-end protection of your critical business information.

System z has been certified at EAL level 5. EAL, or Evaluation Assurance Level, is an industry security certification ranging from level 1 through 7. Only 3 systems have reached EAL 5 (and they are all IBM machines). That has to give you an idea of the strength of System z security. z/OS has achieved EAL 4 certification and DB2 V8 is currently in evaluation at EAL 3 for CAPP and LSPP profiles.

Encryption on the System z platform offers a choice of both standard and optional hardware cryptographic hardware elements. These devices, along with available software components in the z/OS operating system, provide applications with support to invoke the cryptography and to provide key repository management facilities. In this chapter, we focus on the support z/OS provides for cryptography, with specific emphasis on those elements exploited by the Data Encryption for IMS and DB2 Databases.

This chapter contains the following sections:

- ▶ System z integrated cryptography
- ▶ DS8000—Encrypting disk storage
- ▶ TS1120—Encrypting tape storage
- ▶ zIIP

7.1 System z integrated cryptography

The System z platform offers standard and optional hardware cryptographic devices. These devices are available with the proper software components in the different operating systems to provide applications with APIs to invoke the system's hardware cryptography and to provide key repository management facilities. This section contains a description of the cryptographic elements of System z and how they can be invoked by applications.

7.1.1 Cryptographic hardware

IBM has delivered products that implement some of the cryptographic algorithms in hardware (rather than software). The following cryptographic hardware products are available for mainframe servers:

- ▶ Central Processor Assist for Cryptographic Functions (CPACF)
- ▶ Cryptographic Coprocessor Feature (CCF)
- ▶ Peripheral Component Interconnect Cryptographic Coprocessor (PCICC)
- ▶ Peripheral Component Interconnect Cryptographic Accelerator (PCICA)
- ▶ Peripheral Component Interconnect - Extended Cryptographic Coprocessor (PCIXCC)
- ▶ Cryptographic Express2 Coprocessor (CEX2C)
- ▶ Cryptographic Express2 Accelerator (CEX2A)

Table 7-1 shows which of these cryptographic hardware products are available for each of several mainframe servers.

Table 7-1 Cryptographic hardware per server type

Server	CPACF	CCF (feature 0800)	PCICC (feature 0861)	PCICA (feature 0862)	PCIXCC (feature 0868)	CEX2C (feature 0863)	CEX2A (feature 0863)
z800, z900	No	Yes	Yes (requires CCF)	Yes (requires CCF)	No	No	No
z890, z990	Yes (requires z/OS 1.4 or later)	No	No	Yes (requires feature 3863)	Yes (requires feature 3863 and ICSF FMID HCR770A or later)	Yes (requires feature 3863 and z/OS 1.4 or later with ICSF FMID HCR7720 or later)	No
z9 109 z9 BC z9 EC	Yes (requires z/OS 1.6 or later)	No	No	No	No	Yes (requires feature 3863 and z/OS 1.6 or later with ICSF FMID HCR7730 or later)	Yes (requires feature 3863 and z/OS 1.6 or later with ICSF FMID HCR7730 or later)
z10	Yes (requires z/OS 1.8 or later)	No	No	No	No	Yes (requires feature 3863 and z/OS 1.8 or later)	

In the following sections we discuss these cryptographic hardware products.

Attention: For the purpose of this publication, we are restricting our discussion to focus only on the cryptographic capabilities inherent on z9 and z10 and the CEX2C feature. As of the date of this publication, earlier processor classes are out of marketing, and MES upgrades necessary to enable encryption are no longer orderable.

CP Assist for Cryptographic Functions (CPACF)

The CPACF provides the MSA instruction set on every central processor (CP) of a z890, z990, z9 109, z9 BC, and z9 EC server as shown in Figure 7-1.

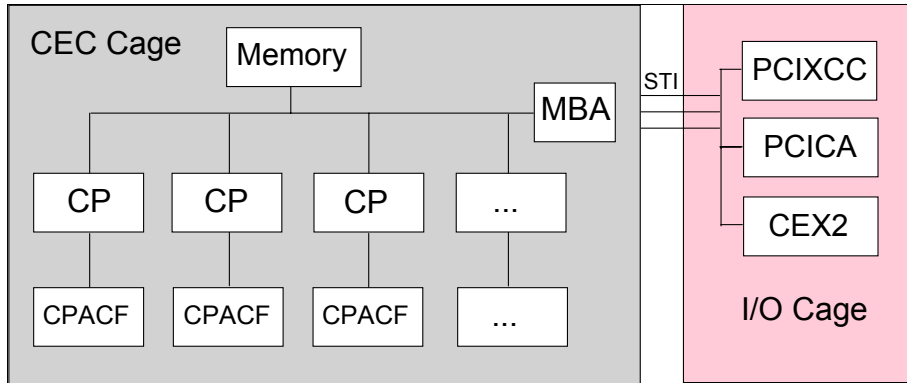


Figure 7-1 A CPACF is associated with every CP

With the DES/TDES Enablement Feature, feature 3863, it also includes the following functions:

- ▶ KM-DEA, KM-TDEA-128, KM-TDEA-192, and KM-AES-128
- ▶ KMC-DEA, KMC-TDEA-128, KMC-TDEA-192, KMC-AES-128, and KMC-PRNG
- ▶ KMAC-DEA, KMAC-TDEA-128, and KMAC-TDEA-192

Important: The CPACF operates with *clear keys* only. A clear key is a key that has not been encrypted under another key and has no additional protection within the cryptographic environment.

Because the CPACF cryptographic functions are implemented in each CP, the potential throughput scales with the number of CPs in the server.

The CPACF hardware that performs the secret key operations (DES, TDES, and AES) and SHA functions operates synchronous to the CP operations. The CP cannot perform any other instruction execution while a CPACF cryptographic operation is being executed.

Crypto Express2 Feature

The CEX2 feature combines the functions of a coprocessor (for secure key encrypted transactions) with the functions of an accelerator (for acceleration of transactions using SSL) into a single optional feature with two PCI-X adapters. Using the HMC console of a z9 system, the PCI-X adapters can be customized to have two coprocessors, two accelerators, or one of each. Figure 7-2 on page 130 shows the layout of a CEX2 feature.

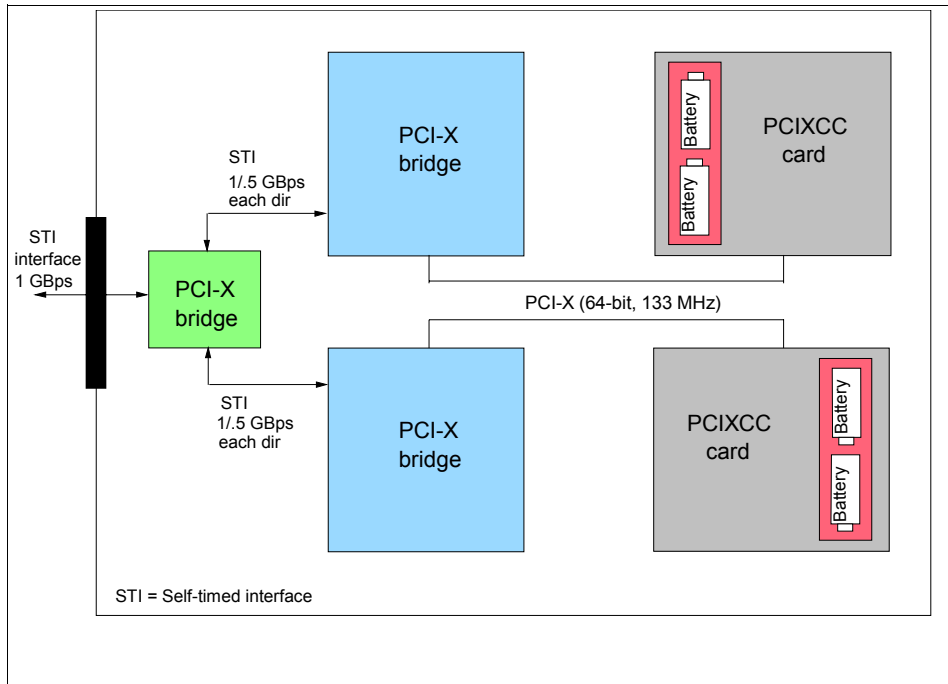


Figure 7-2 Layout of a CEX2 feature

A CEX2 is the generic Crypto Express2 feature. Each engine on the CEX2 can be configured as a coprocessor (CEX2C) or an accelerator (CEX2A). The CEX2-1P is the generic single engine crypto card. It can be configured as a CEX2C-1P or CEX2A-1P.

The CEX2 in coprocessor mode (CEX2C) provides specialized hardware that performs DES, TDES, SHA-1, and RSA operations. The CEX2C is designed to protect the cryptographic keys. Security-relevant cryptographic keys are encrypted under another key (usually a master key) when outside of the secure boundary of the CEX2C card. The master keys are always kept in battery backed-up memory within the tamper-protected boundary of the CEX2C. They are destroyed if the hardware module detects an attempt to penetrate it. The tamper-responding hardware has been certified at the highest level under the FIPS 140-2 standard (Level 4). The CEX2C also supports the clear key PKA operations that are often used to provide SSL protocol communications.

When configured in accelerator mode (CEX2A), the CEX2 feature provides hardware support to accelerate certain cryptographic operations that occur in the e-business environment. Compute-intensive public key operations, as used by the SSL/TLS protocol, can be off-loaded from the CP to the CEX2A, potentially increasing system throughput. The CEX2 in accelerator mode works in clear key mode only.

A z9 109, z9 BC, or z9 EC server can support a maximum of eight CEX2 features. Because each feature provides two coprocessors or accelerators, a z9 server can support a maximum of 16 cryptographic coprocessors or accelerators.

Attention: The minimum number of CEX2C features that can be ordered is two features for each z9 or z10 CEC. The reason for this is to ensure that any cryptographic configuration does not contain a single point of failure, in this case a single CEX2C feature.

The connection of the CEX2 feature to the z9 CPs through the PCI-X bus incurs latency and data transmission time. Because of this connection to the z9 CPs, the CEX2 executes its cryptographic operations asynchronously with a CP operation. For each PCI-X adapter in the CEX2, up to 8 requests can be waiting in the queue either for execution or for dequeuing of the result of a cryptographic operation by a CP. In the CEX2, several operations can be performed in parallel.

Note: In general, cryptographic performance of the Data Encryption for IMS and DB2 Databases is best running under CPACF using clear key.

The CEX2A is actually a CEX2 that has been reconfigured by the user to provide only a subset of the CEX2C functions at enhanced speed. The only functions that remain available when a CEX2 has been reconfigured into a CEX2A are those used for the acceleration of modular arithmetic operations. That is, the RSA cryptographic operations used with the SSL/TLS protocol:

- ▶ PKA decrypt (CSNDPKD) with PKCS 1.2 formatting
- ▶ PKA encrypt (CSNDPKE) with ZERO-PAD formatting
- ▶ Digital signature verification

As shown in Figure 7-2 on page 130, there are two processors installed on a CEX2 feature. As discussed below, these processors can be configured as either both coprocessors, both accelerators, or a mixture of one each. For the z9 or z10 BC customer, a special CEX2C can be ordered. This is referred to as the CEX2C-1P feature code (0870). This special feature code only contains a single processor, configurable either as accelerator or coprocessor. Feature code 0870 can only be installed in a BC class machine. Feature 0863 can be installed in either EC or BC class machines. You need to order two features to provide redundancy.

Verifying Coprocessor configuration

Each PCIXCC has an eight-character serial number and a two-digit adjunct processor (AP) number or ID. The number of APs is limited to 16 on System z9@. A CEX2C is therefore given an AP number between 0 and 15. As shown in Example 7-1, the ICSF Coprocessor Management panel refers to these serial numbers and IDs. The panel prefixes the ID with a letter as follows: A for PCICA, E for CEX2C, F for CEX2A, and X for PCIXCC.

Example 7-1 Sample ICSF Coprocessor Management panel

```
----- ICSF Coprocessor Management ----- Row 1 of 1
COMMAND ==>>                               SCROLL ==>> CSR
```

```
Select the coprocessors to be processed and press ENTER.
Action characters are: A, D, E, K, R and S. See the help panel for details.
```

COPROCESSOR	SERIAL NUMBER	STATUS
-----	-----	-----
E01	94000264	ACTIVE
***** Bottom of data		

Comparison of CPACF, CEX2C, and CEX2A

Table 7-2 summarizes the functions and attributes of the cryptographic hardware available for a System z9.

Table 7-2 Comparison of System z9 cryptographic hardware,

Function or attribute	CPACF	CEX2C	CEX2A
DES/TDES encrypt/decrypt with clear key	X		
AES-128 encrypt/decrypt with clear key	X		
DES/TDES encrypt/decrypt with secure key		X	
Pseudo random number generation (PRNG)	X		
Random number generator (RNG)		X	
Hashing and message authentication	X	X	
RSA functions		X	X
Clear key RSA		X	X
Highest SSL handshake performance			X
Highest performance for asymmetric encryption with clear key			X
Highest performance for asymmetric encryption with secure key		X	
Physically imbedded on each CP	X		
Tamper-resistant hardware packaging		X	
FIPS 140-2 Level 4 certification		X	
ICSF required to be active		X	X
Storage for system master keys		X	
System master keys required to be loaded		X	
Key management operations		X	

Note: A *secure key* is a key that has been encrypted under another key (usually the master key).

7.1.2 IBM Common Cryptographic Architecture

The IBM Common Cryptographic Architecture (CCA) cryptographic API provides the accessibility to cryptographic functions for application programs and administrative software running on the host system. In this section we explain the rationale for the IBM CCA and describe some of its key features.

Rationale for the IBM CCA

The IBM Common Cryptographic Architecture is, as its name implies, an architecture for the use of cryptography. This architecture is common across IBM products and operating system platforms. The IBM CCA is not a product. The IBM CCA was first published in 1990.

Some applications may need to encrypt a file locally and decrypt it either locally or on a remote system at an indeterminate time in the future. This requires that the cryptographic algorithm be compatible in the two systems and that the key used for decryption be available at the remote system.

Other applications may require that communications between systems be enciphered. This requires the use of a common cryptographic key-management and key-exchange approach.

Cryptographic services are tailored for the environment where they will operate. However, they should perform the same operation, with the same results, regardless of the product or the environment. If a customer uses a workstation-based product for user cryptographic services, it should be compatible with host-based products that provide the same services. In addition, if a customer writes an application that requires cryptographic services, the application should be portable between the IBM strategic operating systems. Considering all these requirements, IBM concluded that a cryptographic architecture was both necessary and desirable.

The IBM CCA cryptographic API definition uses a *common key-management approach* and contains a set of *consistent callable services*. A callable service is a routine that receives control when an application program issues a CALL statement.

Common key management ensures that all products that conform to the architecture allow users to share cryptographic keys in a consistent manner. The definition of key management provides methods for initializing keys on systems and networks and also supports methods for the generation, distribution, exchange, and storage of keys.

The callable services provide a common high-level language interface for user or system applications. Thus, an application for a small machine can use the same service calls as an application for a large machine. The services provide a common set of functionality that is applicable to a wide variety of applications. The CCA cryptographic API services define a level of cryptographic capability that allows programs to be developed that work on disparate systems. In particular, the CCA provides two forms of compatibility for applications:

- ▶ Interoperability

Interoperability is the assurance that applications that use the architected services can work together. Interoperability is achieved by using common encryption and decryption algorithms, common key-management definitions, and common external information formats (that is, formats for information sent to another system).

- ▶ Portability.

Portability is the ability to move an application program from one system to another without changing the program. However, it should be noted that portability is at the source code level. That is, it may be necessary to recompile the application program for it to run on a different cryptographic system or to run on a different cryptographic product on the same system.

The objectives in designing the architecture for the CCA cryptographic API were to provide an intuitive multisystem API while allowing for future growth (for example, a new cryptographic algorithm or a new hashing algorithm). Growth can be accommodated by adding new services.

DES key management

Because the DES and TDES algorithms are controlled by keys, the security of protected data depends on the security of the cryptographic key. The CCA can use a master key to protect other keys. Keys are active on a system only when they are encrypted under a variant of the master key, so the master key protects all keys that are used on the system. Keys protected in this manner are also referred to as secure keys.

A master key always remains in a secure area in the cryptographic hardware. In a z/OS environment, an ICSF administrator initializes and changes master keys using the ICSF panels or a Trusted Key Entry (TKE) workstation. All other keys that are encrypted under a master key are stored outside the protected area of the cryptographic hardware. They cannot be attacked because the master key used to encrypt them is secure inside the tamper-protected cryptographic hardware and is zeroed if any attack is attempted. This is an effective way to protect a large number of keys while needing to provide physical security for only a master key.

When the cryptographic hardware is a CEX2C, the master key is called the symmetric key-master key (SYM-MK). A SYM-MK is 192 bits long. However, in the z/OS environment ICSF forces the SYM-MK to be 128 bits long.

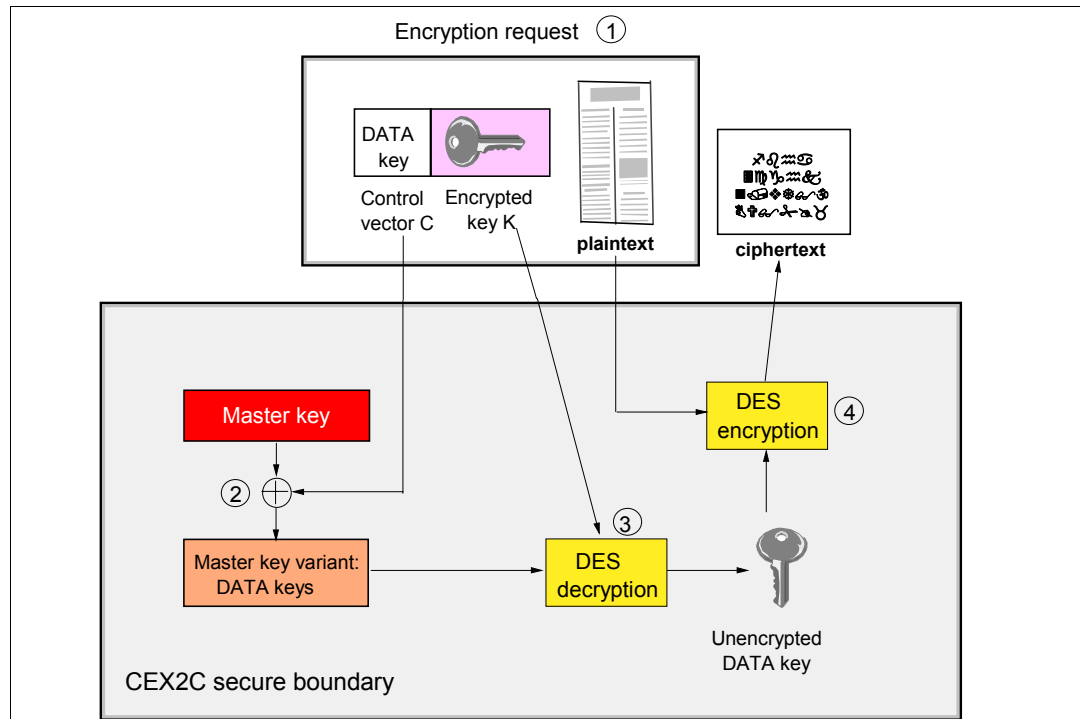


Figure 7-3 Processing inside the CEX2C during an encryption request

► Key label

A key label references a key token, stored in key storage. That token contains the key, either in clear text or in an encrypted format. An example of key storage in the z/OS environment is the ICSF Cryptographic Key Dataset, a data set often called the CKDS. An operational key is a candidate for being kept in key storage if it is a key with a long life, if it is appropriate to control access to this key, or if many users need access to this key.

The `key_identifier` parameter found in most of the cryptographic API callable services allows the programmer to pass keys to the service either directly by value or indirectly through a key label.

Integrated Cryptographic Service Facility

In the z/OS environment, the Integrated Cryptographic Service Facility (ICSF) provides access to cryptographic functions through callable services. The ICSF callable services comply with the IBM CCA cryptographic API and are available for programs written in assembler or high-level languages. IBM CCA supports a hierarchical structure of keys where keys can be encrypted by other keys (key-encrypting keys, KEKs), the master key being at the top of the hierarchy.

ICSF provides cryptographic coprocessors administration facilities for those coprocessors that require a master key to be set.

ICSF also provides key repositories in the form of two VSAM data sets where keys can be kept in key tokens in clear value or encrypted under a KEK or under the coprocessors master keys. The VSAM data sets are the Cryptographic Key Dataset (CKDS) and the Public Key Dataset (PKDS). The key tokens in the CKDS and the PKDS are given a user- or system-defined label that is used for their retrieval and maintenance.

Figure 7-4 is a schematic view of the hardware cryptography implementation in the System z environment. Note that the hardware cryptography technology shown here is the one available on the IBM System z9 and eServer™ zSeries 990 and 890 platforms. The zSeries 800 and 900 host other, although functionally compatible, types of cryptographic coprocessors.

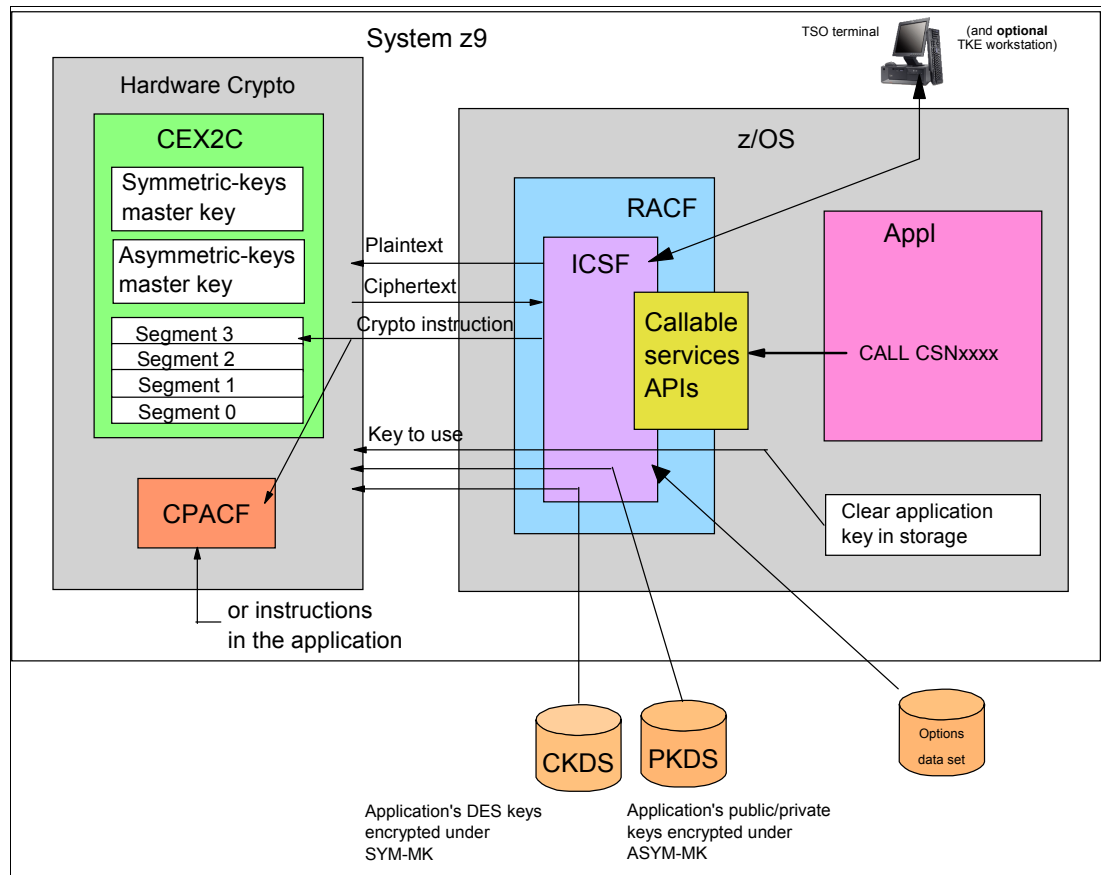


Figure 7-4 Overview of how the hardware and software work together

In Figure 7-4, you can see an application program that has issued a CCA cryptographic API call on a System z9. The call is routed to the ICSF started task. The ICSF started task invokes RACF to determine whether the user ID associated with the request is authorized to

use the requested cryptographic service and any keys associated with the request. If the user ID has the proper authority, the ICSF started task decides whether it should perform the request using ICSF software or cryptographic hardware.

If ICSF decides to use cryptographic hardware, it gives control to its routines that contain the crypto instructions. The crypto instructions that drive the CEX2C are IBM proprietary. IBM does not disclose them.

If ICSF routes the request to the CEX2C and the request is, for example, a request to encrypt data, the ICSF started task provides the CEX2C with the data to be encrypted and the key to be used by the encryption algorithm. Recall that the key is encrypted under a variant of the symmetric key-master key (SYM-MK) stored in the CEX2C. The request proceeds as shown previously in Figure 7-3 on page 134.

The interactions between the functional blocks shown in Figure 7-4 on page 135 are as follows:

- ▶ ICSF is a z/OS started task that offers cryptographic APIs to applications and drives the requests to the Crypto Express2 Coprocessors (CEX2C).
- ▶ The CEX2C is a secure coprocessor in that it contains a master key used to encrypt keys to be kept in storage or in the PKDS data set. The master key resides in the coprocessor hardware only and is used to decrypt, internally to the coprocessor, the secure keys provided so that they can be used to encrypt or decrypt data.
- ▶ ICSF needs other data sets to operate: the CKDS for the use of cryptographic hardware, and an options data set that contains the ICSF started task startup parameters.
- ▶ Installing and maintaining the secret master key is a task that security officers can perform from TSO/E terminals or from an optional Trusted Key Entry (TKE) workstation, the latter for a high security level of the interactions between the security officers and the CEX2C.
If ICSF has access to more than one secure coprocessor, all coprocessors must have been set with the same master key value.
- ▶ The CPACF operates only with clear keys.

The keys can be stored in ICSF-managed VSAM data sets and pointed to by the application program by using the label under which they are stored. The CKDS is used to secure the symmetric keys in their encrypted form and clear keys in their unencrypted form. The PKDS is used to store the asymmetric keys. If the level of ICSF that you are using is HCR7720 or higher, you can also store keys in the CKDS in clear (unencrypted) form.

Controlling who can use cryptographic keys and services

The ICSF administrator can use RACF to control which applications can use specific keys and services. To set up these controls, the ICSF administrator must create RACF general resource profiles in the CSFKEYS resource class and in the CSFSERV resource class. The CSFKEYS class controls access to cryptographic keys, and the CSFSERV class controls access to ICSF services.

The following RACF command defines a profile in the CSFKEYS class:

```
RDEFINE CSFKEYS label UACC(NONE) other-optional-operands
```

IN the above command, *label* is the label by which the key is defined in the CKDS or PKDS. Use the RACF PERMIT command to give user IDs or groups access to the profile:

```
PERMIT label CLASS(CSFKEYS) ID(groupID) ACCESS(READ)
```


To refresh the in-storage RACF profiles, issue a SETROPTS command:

```
SETROPTS RACLIST(CSFKEYS) REFRESH
```

The following RACF command defines a profile in the CSFSERV class:

```
RDEFINE CSFSERV service-name UACC(NONE) other-optional-operands
```

In the above command, *service-name* is chosen from a list in the *z/OS V1R9.0 Cryptographic Services ICSF Administrator's Guide*, SA22-7521. If the application program called the CSNBxxxx service, you should generally specify CSFxxxx as the *service-name* in the RDEFINE command. Note, however, that access to ICSF services CSNBSYE and CSNBSYD is not protected by profiles in the CSFSERV class. Use the RACF PERMIT command to give user IDs or groups access to the profile:

```
PERMIT service-name CLASS(CSFSERV) ID(groupID) ACCESS(READ).
```

To refresh the in-storage RACF profiles, issue a SETROPTS command:

```
SETROPTS RACLIST(CSFSERV) REFRESH
```

ICSF callable services

The format for invoking an ICSF callable service depends on the programming language. For the languages that CICS supports, the formats are as follows:

► C

```
CSNBxxxx (return_code,reason_code,exit_data_length,exit_data,  
parameter_5,parameter_6,...,parameter_N)
```

► COBOL

```
CALL 'CSNBxxxx' USING return_code,reason_code,exit_data_length,  
exit_data,parameter_5,parameter_6,...,parameter_N
```

► PL/I

```
DCL CSNBxxxx ENTRY OPTIONS(ASM);  
CALL CSNBxxxx return_code,reason_code,exit_data_length,exit_data,  
parameter_5,parameter_6,...,parameter_N
```

► Assembler

```
CALL CSNBxxxx,(return_code,reason_code,exit_data_length,exit_data,  
parameter_5,parameter_6,...,parameter_N)
```

You cannot use Java to invoke an ICSF callable service.

Before invoking a callable service in an application program, you must link it into the application program as shown in Example 7-2 on page 138.

Example 7-2 Linking an ICSF callable service into an application program

```
//LKEDENC JOB
/*-----*
/*
/* This JCL links the ICSF encipher callable service, CSNBENC, *
/* into an application called ENCIPHER. *
/* *
/******
//LINK EXEC PGM=IEWL,PARM='XREF,LIST,LET'
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(10,10))
//SYSPRINT DD SYSOUT=*
//SYSLIB DD DSN=CSF.SCSFMODE,DISP=SHR *SERVICES ARE IN HERE
//SYSLMOD DD DSN=MYAPPL.LOAD,DISP=SHR *MY APPLICATION LIBRARY
//SYSLIN DD DSN=MYAPPL.ENCIPHER.OBJ,DISP=SHR *MY ENCIPHER PROGRAM
// DD *
ENTRY ENCIPHER
NAME ENCIPHER(R)
/*
```

Audit trails

ICSF provides SMF records with administrative audit data related to ICSF operations and security officer access to the CEX2C coprocessors. These are the SMF type 82 records.

RACF provides auditing data and violation reports on the CSFSERV and CSFKEYS classes of resources as specified by the RACF auditors. The audit data is provided in SMF records types 80, 81, and 83.

7.1.3 Logical partitioning and System z hardware cryptography exploitation

The functional drawing in Figure 7-4 on page 135 describes cryptographic operations at the logical partition level. Up to 16 different logical partitions can share the same CEX2C card, with two CEX2C cards available in each CEX2C feature plugged in the system.

Note: Each logical partition (LPAR) sharing a physical CEX2C coprocessor is granted a *domain* in the coprocessor. A domain is a set of physical resources that are dedicated to this logical partition. Among the resources provided in a domain are registers intended to securely hold the master keys installed for the ICSF instance that runs in the logical partition.

Domains are part of the CEX2C coprocessor FIPS 140-2 evaluation at level 4, and provide complete isolation and secrecy of individual master keys between the sharing logical partitions.

Domains are manually allocated in the logical partitions image profiles and in the ICSF options data sets.

There is no notion of logical partition sharing with the CPACF because the facility is directly available to any logical partition dispatched on any processing unit.

Introduction to LPAR domains

A cryptographic coprocessor actually has 16 physical sets of master key registers, each set belonging to a domain. A domain is allocated to a logical partition through the definitions in the partition's image profile; the same domain must also be allocated to the ICSF instance running in the logical partition through the options data set.

Each ICSF instance accesses only the master keys corresponding to the domain number specified in the partition's image profile. In other words, each ICSF instance sees a logical crypto coprocessor made of the physical cryptographic engines, shared between all partitions with access to the coprocessor, and of the unique set of registers, the domain dedicated to this partition.

7.1.4 Monitoring the cryptographic workload on z/OS

ICSF collects performance and utilization data for the Crypto Express2 Coprocessors, delivered through SMF types 30, 70, 72, and 82 subtypes 18, 19, and 20 records.

There is no SMF data collection for the utilization of the CPACF.

Tip: For the use of the Data Encryption for IMS and DB2 Databases, an examination of Class 2 CPU will include the time spent in CPACF, but cannot be directly derived as a separate CPU metric.

The data used to monitor the cryptographic services activity is collected and stored into SMF records type 70 subtype 2. This data is collected from ICSF or the Crypto Express2 feature PCIX adapter.

These records are gathered by the RMF Monitor I with the gathering option CRYPTO. They can be processed by the RMF Postprocessor to produce the Crypto Hardware Activity report. The RMF Workload Activity report WLMGL also shows any delays incurred because of unavailable cryptographic devices.

The data shown for CEX2C and CEX2A always reflects the total activity in a system across two types of hardware card:

- ▶ **Crypto Express2 Coprocessor (CEX2C)**

Crypto Express2 provides improved secure key and system throughput. The Crypto Express2 feature supports secure key applications and also offers CVV generation and verification services for 19-digit PANs, providing advanced antifraud security.

The CEX2C coprocessor can be reconfigured in CEX2A.

- ▶ **PCIX Cryptographic Accelerator (CEX2A)**

This cryptographic hardware accelerator card is attached through the PCI bus to the system. This card provides a competitive option to customers that do not need the high security environment of the secure key but need high cryptographic performance for RSA public key algorithms that may be required in Web server applications. The CEX2A provides clear key RSA operations with a modulus length of 1024 bits or 2048 bits. Two algorithms are available, Modulus exponent (ME) and Chinese Remainder Theorem (CRT), for both key lengths.

The data shown for ICSF is related to the partition.

The Crypto Hardware Activity report provides performance measurements on selected ICSF activities:

- ▶ Using the Data Encryption Standard (DES) to encipher and decipher data. DES is probably the best known and most scrutinized encryption algorithm,
- ▶ Generating and verifying message authentication codes (MAC). The MAC is a value calculated from the message according to a secret shared DES key and sent to the receiver together with the message. The receiver can recalculate the MAC and compare it with the MAC received. If the MAC values are identical, the message has not been altered during transmission.
- ▶ Using public hash functions. A hash is calculated from the transmission data according to a public key or function in cases where it is impossible to share a secret key. If the recalculated hash is identical to the one calculated before transmission, data integrity is ensured.
- ▶ Translating and verifying PINs.

7.1.5 Sysplex and System z hardware cryptography

Several instances of ICSF can coexist in a sysplex, with one ICSF instance per sysplex member. They can share the CKDS and PKDS data sets. That is, they share the keys stored in these data sets. Note however the following information:

- ▶ CKDS or PKDS sharing implies having set up the same master keys in the coprocessor domains used by the sharing ICSF instances.
- ▶ Starting with HCR7751, SYSPLEXPKDS sharing is supported, similar to the SYSPLEXCKDS support already available. The SYSPLEXCKDS option in the ICSF installation options data set provides for sysplex-wide consistent updates of the DASD copy of the CKDS and the in-storage copies of the CKDS on all members of the sysplex sharing the same CKDS.

7.1.6 Software requirements

To exploit the enhancements brought to the CPACF (AES, SHA-256, PRNG) on System z9 requires the following specifications, at a minimum:

- ▶ z/OS V1.6 with Cryptographic Support for z/OS V1.6/1.7 and z/OS.e V1.6/1.7 Web deliverable.
- ▶ z/VM V4.4 for guests systems that are to use the CPACF.
- ▶ Linux for System z: Clear key support is available through several Linux distributions.

Exploiting the Crypto Express2 on System z9 when configured as a coprocessor or accelerator (clear or secure key operations) requires the following specifications at a minimum:

- ▶ z/OS V1.6 with Cryptographic Support for z/OS V1.6 and V1.7 and z/OS.e V1.6 and V1.7 Web deliverable.
- ▶ z/VM V5.1 for z/OS and Linux guests, with PTFs
- ▶ z/VM V5.2 for z/OS and Linux guests
- ▶ VSE/ESA V2.7 with PTFs and IBM TCP/IP for VSE/ESA SSL support. z/VSE™ V3.1 with PTFs.
- ▶ Linux on System z - Secure key has been supported since Red Hat® 5.2 and Novell® SUSE® SLES 10 SP 1.

7.1.7 ICSF bibliography

The following guides are available with z/OS:

- ▶ *z/OS ICSF Overview*, SA22-7519
- ▶ *z/OS ICSF System Programmer's Guide*, SA22-7520
- ▶ *z/OS ICSF Application Programmer's Guide*, SA22-7522
- ▶ *z/OS ICSF Administrator's Guide*, SA22-7521
- ▶ *z/OS ICSF Messages*, SA22-7523
- ▶ *z/OS Trusted Key Entry Workstation User's Guide 2000*, SA22-7524

7.2 DS8000—Encrypting disk storage

IBM recognizes the requirement for data protection, not only from hardware or software failures, but also from physical relocation of hardware, theft, and re-tasking of existing hardware. Full Disk Encryption drive sets provide the ability to encrypt data at rest on a DS8000® series storage controller, helping to mitigate the threat of theft, mis-management, or loss of business critical data. The DS8000 series will now have the capability to allow customers to install encrypted 146 GB 15,000 rpm, 300 GB 15,000 rpm, and 450 GB 15,000 rpm Fibre Channel drives with key management services supported by Tivoli Key Lifecycle Manager software (TKLM). The Full Disk Encryption disk drive sets are optional to the DS8000 series.

DS8000 Data Encryption management overview

The Full Disk Encryption disk drive function of DS8000 is depicted in Figure 7-5 on page 142.

- ▶ Using the TKLM (Tivoli Key Lifecycle Manager), the customer configures one or more storage facility images.
- ▶ One or more keys (and associated key labels) are defined within the TKLM
- ▶ One or more TKLM IP ports are defined for each DS8000, up to 4 separate ports are supported, and a minimum of 2 TKLM and TKLM ports are suggested for redundancy
- ▶ One or more encryption groups are configured on the DS8000. The encryption group configuration includes an associated key label. Once the encryption group is configured, the TKLM is contacted, passing the key label. The TKLM performs a key validation and exchange with the DS8000 and passes back one or more keys based on the number of key labels exchanged. This exchange is managed by another component called the Isolated Key Server or SFI. The link between the TKLM and the SFI is through secured SSL.
- ▶ Using the TKLM key(s), the band encryption key, which is set at the factory, is wrapped.
- ▶ The customer then configures an encryption group, which have one or more encryption capable ranks in an extent pool.
- ▶ The DS8000 locks data bands which are located on encrypting disks in extent pools that have been configured into an encryption group.
- ▶ The customer then configures logical volumes in the extent pool. Customer data for these logical volumes are stored on disks with locked data bands
- ▶ If a rank is removed, the disks on the ranks have their wrapped encryption key reset. This also causes all of the data stored on the disks to be erased, this is also referred to as a cryptographic erasure

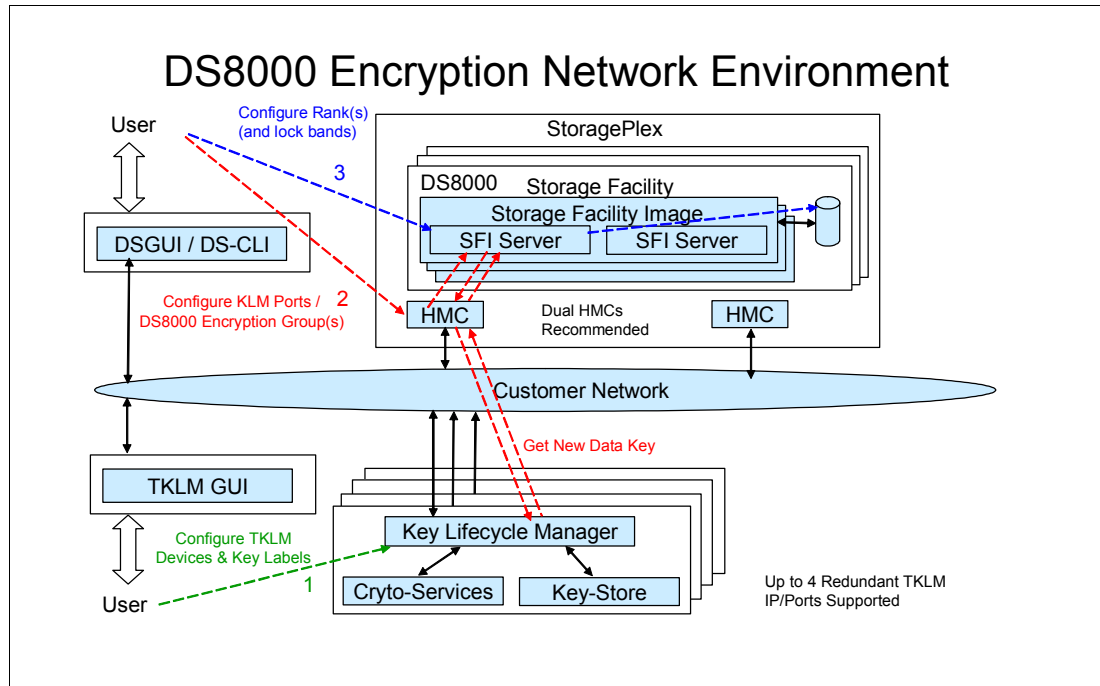


Figure 7-5 DS8000 Encryption Depiction

The Full Disk Encryption support feature is available only as plant order. Plant-configured encryption supporting systems will be allowed to increase the number of drive sets installed at the installed location. Intermixing of drives is not supported, thus the entire subsystem is either encrypted drives or intermixed devices of Fibre Channel, SATA, and SSD devices.

Tivoli Key Lifecycle Manager

Tivoli Key Lifecycle Manager (TKLM) works by allowing administrators to connect to storage devices then create and manage keystores-secure repositories of keys and certificate information used to encrypt and decrypt data-or use existing keystores already in place. Over the course of key lifecycles, all management functions, including creation, importation, distribution, backup, and archiving are accomplished using TKLM's graphic interface, which can be accessed using any standard browser on the network. TKLM thus serves as a central point of control, unifying key management even when different classes of storage devices are involved.

How is the transaction of information between storage devices and TKLM secured? This happens in essentially two stages. First, encryption-capable storage devices are automatically discovered by TKLM and authenticated when initially mounted to ensure that the storage device is actually authorized for that environment. Each storage device generates a pair of RSA keys. TKLM receives and validates these keys using a certificate authority and also confirms through the drive table that the device itself is valid.

Once the device is authenticated, transactions between it and TKLM are secured using a new session key that TKLM generates using the RSA keys. Given this secured communication, an encryption key can then be created for individual cartridges (or virtual cartridges, in the case of a virtual tape library) and sent over the network to the storage device. This sophisticated approach defeats potential security threats, such as rogue device deployment or data interception on the network, by ensuring that every stage in the authentication and encryption process is secure. Yet, the entire process is transparent to the administrator, requiring no oversight and thus reducing operational expenses.

Subsequent key configuration and management is similarly simplified and enhanced. Creating keystores, assigning keys and certificates and managing their total lifecycles is easily achieved within the solution's browser-based GUI. Once TKLM is deployed on a suitable workstation or server, administrators can carry out tasks such as configuration, setup, auditing and compliance support. Key retention policies intended to facilitate compliance initiatives, such as legal discovery, for instance, can be created. Keys can thus be recreated on demand. This feature might also prove useful in cases of disaster recovery by unlocking encrypted backups and thus restoring essential data.

Characteristics of the ES8000 encryption implementation

As currently implemented, there is no mixture of encrypting and non-encrypting disks in an ES8000. For each ES8000, there is a single factory defined key, which is wrapped at the local site with the TKLM managed wrapper. Thus all disks on the single ES8000 share the same key.

Once the ES800 powers up, and connects to the TKLM, there is a key exchange using the keylabel, and if credentials match, then any subsequent access (I/O) to the subsystem retrieves cleartext data.

So, while there is clearly demonstrated protection of the data at rest from attack vectors such as removal of physical media by the CE, or retirement or return of older subsystems, there are other avenues of attack that can justify additional layers of encryption protection. The ES8000 implementation should be viewed as just one element in a layered environment where other encryption solutions can complement the ES8000.

One scenario to describe the requirement for additional defenses is one where the storage administrator has rights to access data at a volume level, but might not have underlying access to the DB2 linear data sets. Because there are catalog privileges granted, the use of tools such as DFSMSdss could allow access to the underlying data, outside of any SQL-based access. A second example would be where a z/OS system programmer brings up another LPAR with little or no RACF protection active. The volumes would be varied offline from the production system, and then online to the sandbox environment, and data could then be compromised in any number of ways without the use of SQL.

In the above scenarios, the ES8000 encryption would provide no protection from nefarious access, but additional encryption techniques, as those provided by the use of the Data Encryption for IMS and DB2 Databases, would further protect the data by encryption implemented at another layer.

7.3 TS1120—Encrypting tape storage

The IBM System Storage™ TS1120 Tape Drive has been the first drive to be enhanced to provide the customer the option of using drive-based data encryption.

IBM now has two tape drives that are capable of encrypting the data written on the tape cartridge. They are the IBM System Storage TS1120 Model E05 Tape Drive and the IBM Systems Storage Linear Tape-Open (LTO) Ultrium Generation 4 Tape Drive. In this section we refer to them as the TS1120 tape drive and the LTO4 tape drive.

While other encryption solutions require hardware resources or processor power when using software encryption, Tape Encryption is done with little or no impact on the performance of the TS1120 tape drive or the LTO4 tape drive. You can easily exchange encrypted tapes with your business partners or data centers that have the necessary key information to decrypt the data.

With the original encryption announcement for the TS1120 tape drive, IBM also introduced a new IBM Encryption Key Manager component for the Java Platform feature (Encryption Key Manager or EKM) that is designed to generate and communicate encryption keys for tape drives across the enterprise. The new feature uses standard key repositories on supported platforms. The IBM Tape Encryption solution provides an enterprise key management solution with common software for Open Systems and mainframe environments that allow sharing of a common keystore across platforms. Integration with z/OS policy, key management, and security capabilities provides a proven, highly secure infrastructure for encryption key management.

IBM tape encryption provides high performance data encryption. Encryption is performed at the tape drive hardware at a speed of up to 120 Mbps (for uncompressed data) and even higher speeds for data that compresses. It also supports encryption of large amounts of tape data for backup and archive purposes.

The IBM Tape Encryption solution, utilizing the TS1120 Tape Drive or the LTO4 Tape Drive, offers a cost-effective solution for tape data encryption by offloading encryption tasks from the servers, leveraging existing tape infrastructure incorporated in standard IBM Tape Libraries, and eliminating the need for unique appliance hardware.

How tape encryption works

Encryption, implemented in the tape drive, encrypts the data before it is written to the cartridge. When tape compression is enabled, the tape drive first compresses the data to be written and then encrypts it. This means that there is no loss of capacity with IBM tape encryption. If the encryption solution encrypts the data first and then tries to compress the data, the encrypted data will usually compress little if at all.

To encrypt the data, the tape drive needs a key. The key is provided by the Encryption Key Manager, and it is provided in an encrypted form to make the tape encryption solution secure.

Key management can be handled either internally by an application, such as Tivoli Storage Manager, or externally by an Encryption Key Manager.

The IBM Encryption Key Manager (EKM) component for the Java platform is a Java software program that assists IBM encryption-enabled TS1120 tape drives and Linear Tape-Open (LTO) Ultrium 4 tape drives by providing, protecting, storing, and maintaining encryption keys that are used to encrypt information being written to, and decrypt information being read from, tape media. EKM operates on a variety of operating systems. EKM is designed to be a shared resource deployed in several locations within an enterprise. It is capable of serving numerous IBM encrypting tape drives regardless of where those drives reside (for example, in tape library subsystems, connected to mainframe systems through various types of channel connections, or installed in other computing systems). IBM supplies the EKM free of charge.

EKM acts as a process awaiting key generation or key retrieval requests sent to it through a TCP/IP communication path between EKM and the tape library, tape controller, tape subsystem, device driver, or tape drive. When a tape drive writes encrypted data, it first requests an encryption key from EKM. The tasks that the EKM performs upon receipt of the request are different for the TS1120 tape drive and the TS1040 tape drive

Figure 7-6 on page 145 summarizes the process flow for tape encryption using TS1120.

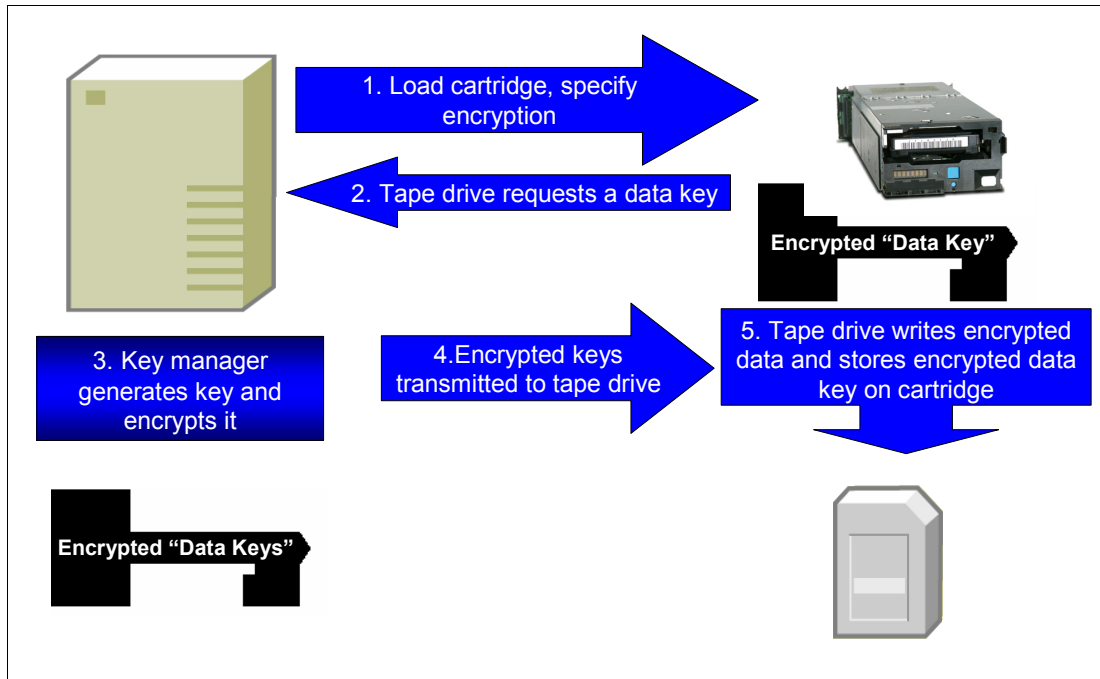


Figure 7-6 TS1120 tape encryption process flow

EKM requests an Advanced Encryption Standard (AES) key from the cryptographic services and serves it to the tape drives in two protected forms:

- ▶ Encrypted or wrapped, using Rivest-Shamir-Adleman (RSA) key pairs. TS1120 tape drives write this copy of the key to the cartridge memory and three additional places on the tape media in the cartridge for redundancy.
- ▶ Separately wrapped for secure transfer to the tape drive where it is unwrapped upon arrival and the key inside is used to encrypt the data being written to tape.

When an encrypted tape cartridge is read by a TS1120 tape drive, the protected AES key on the tape is sent to EKM where the wrapped AES key is unwrapped. The AES key is then wrapped with a different key for secure transfer back to the tape drive, where it is unwrapped and used to decrypt the data stored on the tape. EKM also allows protected AES keys to be rewrapped, or rekeyed, using different RSA keys from the original keys that were used when the tape was written. Re-keying is useful when an unexpected need arises to export volumes to business partners whose public keys were not included. It eliminates the need to rewrite the entire tape and enables a tape cartridge's data key to be re-encrypted with a public key.

Figure 7-7 on page 146 summarizes the LTO4 tape encryption process flow.

The EKM fetches an existing AES key from a keystore and wraps it for secure transfer to the tape drive where it is unwrapped upon arrival and used to encrypt the data being written to tape.

When an encrypted tape is read by an LTO Ultrium 4 tape drive, the EKM fetches the required key from the keystore, based on the information in the Key ID on the tape, and serves it to the tape drive wrapped for secure transfer.

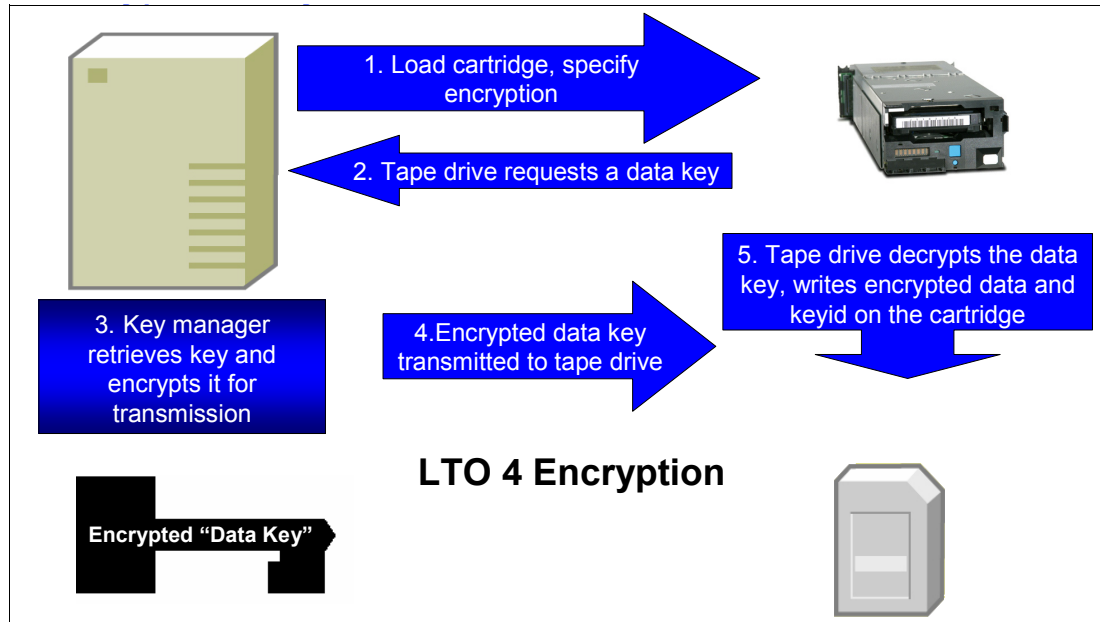


Figure 7-7 LTO4 Tape Encryption process

What to encrypt

The loss of computer backup tapes is one type of event that triggers consumer notification. This has led to increasing data protection requirements.

What should you encrypt, and just as importantly, what should you not encrypt? There is an ever increasing focus on data security:

- ▶ California is generally considered the first state to implement a law requiring disclosure of security breaches in July, 2003.
- ▶ Legislation has been enacted by 22 states that requires notification in cases of security breaches. See the following Web page for more information:
<http://www.Privacyrights.org>
- ▶ Similar federal legislation has been proposed. See the following Web page for more information:
http://www.epic.org/privacy/bill_track.html

Data protection requirements are driven by a variety of reasons. In addition to regulatory requirements that are driving the need for greater data security, integrity, retention, auditability, and privacy, reasons for increasing data protection are as follows:

- ▶ Severe business impacts that might be caused by loss or theft of data, including financial liability, reputation damage, legal risk, and compliance risk
- ▶ Increasing need to share data securely with IBM Business Partners and maintain backups at remote locations
- ▶ Need to reduce complexity and improve processes around enterprise encryption management
- ▶ Requirement to cost-effectively encrypt large quantities of tape data

There are additional drivers for encryption in financial services:

- ▶ A riskier environment

Internet banking, for example, relies on open networks with multiple access points to conduct business in real time to drive down costs and improve response times to revenue generating opportunities.

- ▶ Growing regulatory burden, such as the following legislation:

- Gramm-Leach-Bliley Act (GLBA) of 1999
- California Law No. SB 1386
- FCRA/FACTA amendments
- Basel II

Not all of the regulations specifically require the use of stored data encryption. However, many organizations are implementing encryption for their protected information in conjunction with other security layers to protect personally-identifiable information.

- ▶ Maturing industry standards, such as the Payment Card Industry (PCI) Data Security Standard (DSS)

Why use tape encryption

Tape encryption is used to hide and protect sensitive data. If tape data on cartridges leaves data centers, the data is no longer protected through RACF or similar access protection mechanisms. Tape encryption can help fulfill security regulations. Many governmental agencies require disclosure of security breaches. Industry organizations are also increasing their scrutiny of security procedures. Now, tape encryption uses an easy and economical way to protect data from unauthorized view.

Important and sensitive data can be protected in many ways. Data can be encrypted by means of special software programs, hardware adapters, facilities, or outside of the device where the data is stored. Encrypting data with software programs takes away processor power, and encrypting data with hardware requires additional investment in hardware for the computers.

The advantage of IBM tape encryption is that the data is encrypted after compression, and there are no additional software program costs. IBM Tape Encryption saves space on tape cartridges and saves additional hardware investments. In addition, outboard encryption in the tape drives might help you protect large volumes of tape data in a cost-effective way. Data on cartridges does not have to be degaussed or overwritten with patterns of x'FF' at the end of life of the cartridge. This is valid for Write Once Read Many (WORM) cartridges and normal cartridges.

The new encryption key management capability is designed to manage keys across mainframes and open systems environments. There is only one component to be managed across multiple platforms.

Tape encryption can be managed by the applications or can be system-managed or library-managed.

Why encrypt data in the drive

The IBM tape-drive encryption solution encrypts the data within the drive using the 256-bit AES algorithm, rather than receiving previously encrypted data. There are several advantages to this system. By encrypting data in the drive, the drive can offer the most efficient data compression, because the drive first compresses the data, then encrypts it, providing more efficient data storage and media usage.

Encrypting in the drive also eliminates the need for any additional machines or appliances in the environment by offloading the encryption processing overhead onto the drive. Because the drive can also process unencrypted workloads, the IT environment is further simplified, eliminating the need for separate drives to process data that does not need to be encrypted.

Summary

Encryption capability that is provided as a standard feature in the IBM TS1120 tape drive or the IBM LTO4 tape drive makes encrypting data stored on tape cartridges much easier. This is increasingly important as legislation continues to grow requiring notifying individuals when their personal information has potentially been compromised. The IBM-developed tape drive-based encryption solutions described here, coupled with the new EKM component, enable key management and encryption in a wide variety of environments.

IBM provides tape drive encryption support in a wide range of operating systems environments:

- ▶ z/OS
- ▶ z/VM
- ▶ z/VSE
- ▶ z/TPF
- ▶ i5/OS®
- ▶ AIX®
- ▶ Linux on System z
- ▶ Linux on other platforms
- ▶ HP-UX
- ▶ Sun™ Solaris™
- ▶ Windows Server® 2000 or Windows 2003 Server

For more information about tape encryption, see *IBM System Storage Tape Encryption Solutions*, SG24-7320, *IBM Virtualization Engine TS7500: Planning, Implementation, and Usage Guide*, SG24-7520 and the white paper by the Enterprise Strategy Group available from the following Web page:

http://www.ibm.com/servers/eserver/zseries/zos/pdf/White_Paper_ESG_System_z_final.pdf

7.4 zIIP

In this section we examine the use of zIIP specialty engine with encryption.

7.4.1 IPsec encryption and zIIP exploitation

IPsec is an open networking standard used to create highly secure connections between two points in an enterprise. This may be server-to-server, or server-to-network device, as long as they support the IPsec standard. Using IPsec to provide end-to-end encryption helps provide a highly secure exchange of network traffic, and can help satisfy regulatory requirements.

The z/OS Communications Server allows portions of IPsec processing to take advantage of zIIPs. The zIIP Assisted IPsec function moves a portion of the IPsec processing from the general purpose processors to the zIIPs. With zIIP Assisted IPsec, the zIIPs, in effect, become an encryption engine. In addition to performing the encryption processing, the zIIP will also handle cryptographic validation of message integrity, and IPsec header processing. This capability was available August 2007 through z/OS V1.8 and PTFs and native in z/OS V1.9.

z/OS Communications Server can direct CPU-intensive IPsec processing to an IBM System z9 Integrated Information Processor (zIIP). This can lower the computing cost incurred by the IPsec protocols, while at the same time increasing the processing capacity of general purpose CPs.

This function can also be enabled on machines with no zIIPs, so that you can project the effectiveness of zIIP for your current IPsec workload. When this function is enabled on a z/OS server with no zIIPs, z/OS accounts for the zIIP-eligible workload that was processed on CPs, in SMF record types 30 and 7x. You can use this accounting information to project the percentage of your workload that would be zIIP-eligible, if you had zIIPs configured to your MVS image. Information concerning performance measurements of IPsec and zIIP, along with help with workload sizing and capacity planning for this capability, reference the whitepaper available from the following Web page:

<http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100988>

7.4.2 zIIP and Encryption Tool for IMS and DB2 Databases

The zIIPs are available on the z9 / z10 EC and BC servers. zIIPs are used to offload specialized subsystems related processing elements from z/OS configured as enclave SRBs. These elements are initially associated with some types of DB2 processing.

DB2 V8 moves eligible DRDA, selected DB2 utility (most index management work), part of Business Intelligence (BI) workloads to a zIIP, reducing software cost and improving available capacity of existing general purpose engines.

DB2 9 adds remote native SQL procedures execution and more instances of eligible query parallelism (BI) work. Furthermore, z/OS XML System Services processing executes on the zAAP and DB2 can direct the full amount of the z/OS XML System Services processing to zIIPs when it is used as part of any zIIP eligible workload (like DRDA).

As discussed earlier, CPACF can execute on any available CP or IFL engine. So, for zIIP eligible workload accessing encrypted tables, some of the overhead of CPACF-supported encryption could be offloaded to zIIP.

Clear key encryption is done on the CPACF. Secure key encryption is done on the CEX2C.



z/OS security

In this chapter we provide some background information about various components of the z/OS operating system that play a role in providing a robust security framework.

We discuss security elements of the following z/OS operating system components:

- ▶ Integrated Cryptographic Service Facility
- ▶ Communication Server
- ▶ z/OS Encryption Facility

8.1 Integrated Cryptographic Service Facility

Integrated Cryptographic Service Facility (ICSF) is the component of the z/OS operating system that provides cryptographic support through the use of hardware and software implementations. In conjunction with System z hardware enablement, cryptographic support can be extended to a number of subsystem and middleware environments on the System z platform.

Exploitation of the various cryptographic features of ICSF can be achieved through the use of either CPACF (cryptographic capability on each general purpose or speciality processor) or CEX2 (optional hardware for secure key encryption operations).

8.1.1 Middleware ICSF exploitation

The major middleware products exploiting ICSF briefly described here are CICS, WebSphere, IMS, and VSAM. We discuss DB2 for z/OS in detail in 3.2, “DB2” on page 51.

CICS

CICS provides the capability for the CICS application programmer to code calls to the encryption services using the ICSF API. When operating in a multi-tasking environment such as CICS, there are some special considerations when implementing an ICSF exploitation. CICS application programs that wish to use the ICSF API should also use the CICS-ICSF Attachment Facility.

The purpose of the CICS-ICSF Attachment Facility is to enhance the performance of CICS transactions in the same region as a transaction using long-running ICSF services such as the PKA services and CKDS or PKDS update services. You must have CICS 4.1 or higher.

Without the CICS-ICSF Attachment Facility, the application that requests a long-running ICSF service is placed into an OS WAIT. This affects any other transactions that run in the same region. The CICS-ICSF Attachment Facility consists, in part, of a CICS Task-Related User Exit (TRUE). The TRUE attaches a task control block (TCB, which makes the call to the ICSF service. This allows the CICS application that requests the long-running service to be placed into a CICS WAIT, rather than an OS WAIT, for the duration of the operation.

Before you can use the CICS-ICSF Attachment Facility, the ICSF system programmer, or the CICS administrator needs to install it.

WebSphere

WebSphere Application Server supports the use of several cryptographic functions in conjunction with Web services. It implements the Web services security (WS-Security) version 1.0 standard, including digital signature, encryption, and security tokens. It supports transport layer security, HTTP basic authentication, and message level security.

The IBM mainframe hardware can execute many cryptographic functions and allow cryptographic functions to execute in hardware designed specifically for that purpose. This results in significantly improved performance for specific execution paths.

The WebSphere Application Server support for Web services security uses the Java Cryptography Extension (JCE) framework to perform cryptographic functions. This framework provides a link between implementation-agnostic users of cryptographic functions and one or more implementations, which may exhibit unique or idiosyncratic characteristics and optimizations. Which implementations are used by a given installation is determined by a list of providers found in the file `java.security`.

When the JVM™ is initialized, it interrogates each provider in the list to determine which cryptographic functions the provider is able to perform. Subsequently, when a cryptographic function is invoked by a user, the JVM will delegate the function to the first provider in the list capable of performing that function. The provider that is able to invoke System z cryptographic hardware is IBMJCECCA described on the following Web page:

<http://www.ibm.com/servers/eserver/zseries/software/java/products/j5jcecca.html>

IMS

While not covered in detail in this publication, the data encryption for IMS and DB2 databases can encrypt IMS database data in a fashion similar to the DB2 implementation described in this publication. Encryption and decryption follow a unique flow of processing in the IMS environment, and the IMS environment poses several requirements and considerations.

The IMS application program passes a segment REPL, ISRT, or LOAD request to the IMS control region. IMS uses the DBD to determine that a segment edit/compression exit is required, so IMS loads the exit. The exit invokes ICSF services, passing the user-defined data encryption key label (provided by the exit) and the unencrypted segment. The key label refers to a DATA, CLRDES, or CLRAES key type that must be predefined by the ICSF administrator. When the segment has been successfully encrypted, the exit passes the segment back to IMS. IMS puts the encrypted segment into the database.

Similar to the requests described above, when a GET request is passed to the IMS control region, IMS determines, from the DBD, that a segment edit/compression exit is required, so IMS loads the exit. IMS retrieves the encrypted segment from the database. IMS then calls the exit and passes it the encrypted segment. The exit invokes ICSF services, which passes the user-defined data encryption key label (provided by the exit) and the encrypted segment. The key label refers to a DATA, CLRDES, or CLRAES key type that must be predefined by the ICSF administrator. When the segment has been successfully decrypted, the exit passes the segment back to IMS. IMS passes the decrypted segment back to the application.

There are some restrictions covering the use of the Data Encryption for IMS and DB2 Databases in an IMS environment:

- ▶ You cannot both encrypt and compress the database.
- ▶ An IMS segment can be associated with only one segment edit/compression exit. If your IMS segment is already associated with a segment edit/compression exit and you want to implement Data Encryption for IMS and DB2 Databases, you must code an alternative solution for your existing exit.
- ▶ HIDAM index databases cannot be encrypted (the IMS DBD COMPRTN parameter does not allow index databases to be specified on the segment edit/compression exit).

There are also some special considerations when evaluating the use of the Data Encryption for IMS and DB2 Databases in an IMS environment:

- ▶ Depending on your security requirements, you can define encryption key labels for as many segments as you need. (Encryption key labels are set up by your security analyst.)
- ▶ An exit must be built for each encryption key label that you define. Note that you need to balance your security requirements against the increased maintenance of multiple exits.
- ▶ The first time that you use segment edit/compression exits at your installation, your system programmer needs to provide APF authorization for the segment edit/compression EXITLIB.
- ▶ If you are already using segment edit/compression exits, you need to ensure that the segment edit/compression exits reside in an APF-authorized EXITLIB.

For further information about implementing Data Encryption for IMS and DB2 Databases in an IMS environment, refer to *IBM Encryption Tool for IMS and DB2 Databases User Guide*, SC18-9549.

VSAM encryption

There is limited support for VSAM encryption and decryption available from an ICSF perspective. The IDCAMS REPRO facility provides some limited support with the use of the IDCAMS ENCRYPT and DECRYPT options. These options only apply to the use of IDCAMS REPRO and are generally only useful when needing to encrypt a REPRO output file to share with a partner outside the physical datacenter boundaries. This approach will not provide any mechanism for key sequenced access initiated by applications such as CICS file control requests for read and browse operations.

Some restrictions affect the use of the IDCAMS encryption facility:

- ▶ The length of the data encryption key is limited to 8 bytes, or 56-bit DES. Triple DES support is not available. As mentioned elsewhere, 56-bit DES is now considered a weak encryption algorithm and is generally not recommended for situations which require a demonstration of strong encryption capabilities.
- ▶ Key labels are limited to 8 characters because of the fixed size of REPRO storage areas.
- ▶ The REPRO command's encryption algorithm variables are not documented, so you cannot use them to write decryption applications on another system. Therefore, cross-platform exchange is not possible.

VSAM encryption requires the definition of a key value that is used to encrypt and decrypt the data key. To define the key value, use the ICSF key administrative KGUP utility. Also, ensure that ICSF can support PCF macro calls by specifying COMPAT(YES) in the ICSF installation options.

If there exists a strong requirement to encrypt VSAM data at rest, a better solution might be to implement an encryption solution that uses the DS8000 encrypting disk technology.

8.1.2 Resource Access Control Facility

Resource Access Control Facility (RACF) is a security program. It is a component of the Security Server for z/OS. RACF controls what you can do on the z/OS operating system. You can use RACF to protect your resources. RACF protects information and other resources by controlling the access to those resources. RACF provides security through the following methods:

- ▶ Identifying and verifying users
- ▶ Authorizing users to access protected resources
- ▶ Recording and reporting access attempts

Access control

RACF identifies you when you log on to the operating system you want to use. It does so by requiring a user identification, the user ID (a unique identification string). RACF then verifies that you are the user you say you are by requesting and checking a password. Each RACF user ID has a unique password. You should be the only one who knows your password. That way, RACF can ensure personal accountability.

RACF enables your organization to define individuals and groups who use the system RACF protects. For example, for a secretary in your organization, a security administrator uses RACF to define a user profile that defines the secretary's user ID, initial password, and other information.

A group is a collection of individuals who have common needs and requirements. For example, the secretaries for a whole department might be defined as one group. RACF also enables an installation to define what authorities you have, or what authorities possessed by a group to which you belong. RACF controls what you can do on the system. Some individuals have a great degree of authority, while others have little authority. The degree of authority you are given is based on what you need to do your job.

Besides defining user and group authorities, RACF protects resources. A resource is your organization's information stored in its computer system, such as a data set. For example, a secretary might have a data set as a resource.

RACF provides a way to control who has authority to access a resource. RACF stores information about users, groups, and resources in profiles. A profile is a record of RACF information that has been defined by the security administrator. There are user, group, and resource profiles.

Using information in its profiles, RACF authorizes access to certain resources. RACF applies user attributes, group authorities, and resource authorities to control use of the system.

- ▶ Your user profile provides your user attributes. User attributes describe what system-wide and group-wide access privileges you have to protected resources.
- ▶ Your group profile describes the kind of authority you as a group member have to access resources that belong to your group.
- ▶ The resources have profiles describing the type of authority needed to use them.

The security administrator, or someone in authority in your organization, controls the information in your user profile, in group profiles, and in resource profiles. You, as the user, control the information in profiles describing your own resources, such as your own data sets. You can protect your data by setting up resource profiles.

A resource profile can contain an access list and a default level of access authority for the resources it protects. An access list identifies the access authorities of specific users and groups, while the default level of access authority applies to anyone not specifically in the access list. You can specify the users you want on the access list and what authority they have to use your data. You can change your resource profiles, but you cannot change the user or group profiles, because they are established by the system administrator.

RACF can protect the following types of general resources:

- ▶ Disk volumes
- ▶ Tape volumes
- ▶ Load modules (programs)
- ▶ Application resources (such as resources for IMS, CICS, and DB2)
- ▶ Terminals
- ▶ Installation-defined resources

Resources are protected with profiles. A profile contains descriptive information about a user, a group, or resource. RACF uses the information in a profile to control use of protected resources. When you attempt to use a protected resource, RACF checks your user profile, and the resource profile, to decide whether to allow you to use the resource.

Audit reporting

Besides uniquely identifying and authorizing you, RACF can record what you do on the system. It keeps track of what happens on the system so that your organization can monitor who is logged on the system at any time. RACF reports if persons have attempted to perform

unauthorized actions. For example, RACF can record when someone who does not have the proper authority tries to use or change your data.

RACF provides the following assistance to help you to audit access control and accountability:

- ▶ Logging routines that record the information you require
- ▶ Audit control functions that enable you to specify the information RACF is to record (or log).
- ▶ The RACF SMF data unload utility, which converts SMF records into formats which can be used by a relational database manager, such as an XML version that can be easily viewed by a web browser.
- ▶ The DFSORT ICETOOL, which generates reports from RACF SMF data unload utility information and RACF database unload utility information.
- ▶ The data security monitor (DSMON), which generates reports containing information about the security environment for MVS.
- ▶ The RACF report writer, which generates tailored reports based on the information you have directed RACF to log.

After RACF has logged security events, you can analyze this log through the following techniques:

- ▶ Loading the records produced by the RACF SMF data unload utility into a relational database manager for analysis.
- ▶ Creating XML output of the report and viewing the results in a web browser. This report can also be customized by the use of an XSLT stylesheet file.
- ▶ Invoking the RACF report writer to print out the data RACF has logged and use the reports to identify possible security violations or weaknesses in the security mechanism.

Logging, the recording of data about specific events, is the key to auditing the use of RACF at your installation. You must ensure that RACF logs the information you need. RACF uses the system management facilities (SMF) to log data about various RACF events. RACF writes SMF records to an SMF data set or log stream.

RACF audit data is a record of an installation's security relevant events. This data is used to verify the effectiveness of an installation's security policy, determine whether the installation's security objectives are being met, and identify unexpected security relevant events. The RACF SMF data unload utility (IRRADU00) enables installations to create a sequential file from the security relevant audit data. The sequential file can be used in several ways:

- ▶ Viewed directly
- ▶ Used as input for installation-written programs, manipulated with sort/merge utilities
- ▶ Output to an XML-formatted file for viewing on a web browser
- ▶ Uploaded to a database manager (for example, DB2) to process complex inquiries and create installation-tailored reports

It is not intended to be used directly as input to RACF commands.

The output file from the RACF SMF data unload utility can one of the following formats:

- ▶ Viewed directly
- ▶ Used as input to your own programs
- ▶ Manipulated with sort/merge utilities
- ▶ Used as input to a database management system so you can produce reports tailored to your requirements
- ▶ Viewed using a web browser

8.2 Communication Server

z/OS Communications Server is a network communication access method. It provides both Systems Network Architecture (SNA) and Transmission Control Protocol/Internet Protocol (TCP/IP) networking protocols for z/OS.

The TCP/IP protocol suite (also called stack), includes associated applications, transport- and network-protocol layers, and connectivity and gateway functions. TCP/IP is a set of protocols and applications that enable you to perform certain computer functions in a similar manner independent of the types of computers or networks being used.

When you use TCP/IP, you are using a network of computers to communicate with other users, share data with each other, and share the processing resources of the computers connected to the TCP/IP network.

A computer network is a group of computer nodes electronically connected by some communication medium. Each node has the hardware and the programs necessary to communicate with other computer nodes across this communication medium. The node can be a PC, workstation, departmental computer, or large computer system. The size of the computer is not important. The ability to communicate with other nodes is important. Computer networks enable you to share the data and computing resources of many computers. Applications, such as departmental file servers, rely on networking as a way to share data and programs.

Many forms of communication media are available today. Each is designed to take advantage of the environment in which it operates. Communication media consist of a combination of the physical network used to connect the computer nodes and the language, or protocol, they use to communicate with each other.

z/OS Communications Server commands and TCP/IP provide a set of basic functions:

- ▶ Logging on to other hosts
- ▶ Transferring data sets and files between hosts
- ▶ Sending and receiving mail
- ▶ Using other hosts
- ▶ Printing to or from other hosts

The SNA protocols are provided by VTAM® and include Subarea, Advanced Peer-to-Peer Networking (APPN), and High Performance Routing protocols. z/OS Communications Server provides the interface between application programs residing in a host processor, and resources residing in an SNA network. It also links peer users in the network.

Secure Sockets Layer

Secure Sockets Layer (SSL) is a communications protocol that provides secure communications over an open communications network (for example, the Internet). The SSL protocol is a layered protocol that is intended to be used on top of a reliable transport, such as Transmission Control Protocol (TCP/IP). SSL provides data privacy and integrity and server and client authentication based on public key certificates. Once an SSL connection is established between a client and server, data communications between client and server are transparent to the encryption and integrity added by the SSL protocol.

System SSL uses the Integrated Cryptographic Service Facility (ICSF) if it is available. ICSF provides hardware cryptographic support that is used instead of the System SSL software algorithms. System SSL also takes advantage of the CP Assist for Cryptographic Function (CPACF) when available. System SSL checks for the hardware support during its runtime initialization processing and will use the hardware support if available. For System SSL to use

the hardware support provided through ICSF, the ICSF started task must be running prior to the application and the application user ID must be authorized to the appropriate resources in the RACF CSFSERV.

System SSL handshake processing uses both the RSA and digital signature functions that are expensive functions when performed in software. For installations that have high volumes of SSL handshake processing, utilizing the capabilities of the hardware will provide maximum performance and throughput. For example, on a z9 EC, z9 BC and System z10™ EC, having both CEX2C and CEX2A will result in the maximum clear key RSA and digital signature processing being done in hardware.

z/OS Communications Server and Security

A set of protections are built into z/OS and z/OS Communications Server to enhance the security level of the z/OS connection to a non-secure network by preventing network-issued attacks from affecting applications running in z/OS. These protections focus on detecting such attacks and alerting the system's operations team if they occur. Built-in protections are also intended to prevent the z/OS instance from being an initiator or participant in an attack aimed at another system in the network.

- ▶ IP filtering allows a network administrator to control the network traffic into or out of the z/OS by selectively denying access to specific IP packets based on corporate security policy.
- ▶ IPSec provides authentication, integrity, and data privacy between any two TCP/IP stacks. It creates a virtual private network (VPN) enabling an enterprise to extend its local and private network across a public network, for the secure transfer of data.
- ▶ IPSec in z/OS exploits the hardware cryptographic capabilities of the System z.z/OS.
- ▶ Application Transparent Transport layer Security (AT-TLS) performs the SSL or TLS protocol in the TCP layer of the stack transparently to the applications that use TCP/IP sockets. This reduces or eliminates application development overhead, maintenance, and parameter specification, because applications do not need to be designed to support SSL or TLS. Note that AT-TLS also provides an API that AT-TLS-aware applications can use to interact with the internal processes of the SSL/TLS protocol.
- ▶ Intrusion Detection Services (IDS) detects, records, and defends against TCP/IP ports scans, IP datagram-based attacks, flooding, and denial of service attacks.
- ▶ Protection of network resources is performed by RACF through the SAF interface on the z/OS platform. RACF protects network resources such as the TCP/IP stack, TCP/IP ports, network management commands, and resources from unauthorized access.

IPSec or AT-TLS

IPSec is intended to protect traffic between two TCP/IP stacks, absolutely transparently to the applications, and can potentially protect any TCP/IP protocol that TCP/IP datagrams are carrying. It is an industry-wide adapted protocol, and you can expect interoperability between different platforms.

Beginning with z/OS V1R9, or z/OS V1R8 with APARs PK40178 and OA20045, users of z9 systems with one or several zIIP specialty engines installed can move most of the IPSec processing from the general purpose processors to the zIIPs. This saves CP MIPS that would have otherwise been consumed. This offloading to zIIPs is optional and is selected using a configuration statement in the TCPIP.PROFILE that triggers the Communications Server to request z/OS to direct the IPSec enclave SRB processing to available zIIP. Note that the IPSec workload is competing for zIIP cycles with other zIIP-eligible workloads (such as DRDA processes). See Figure 8-1 on page 159.

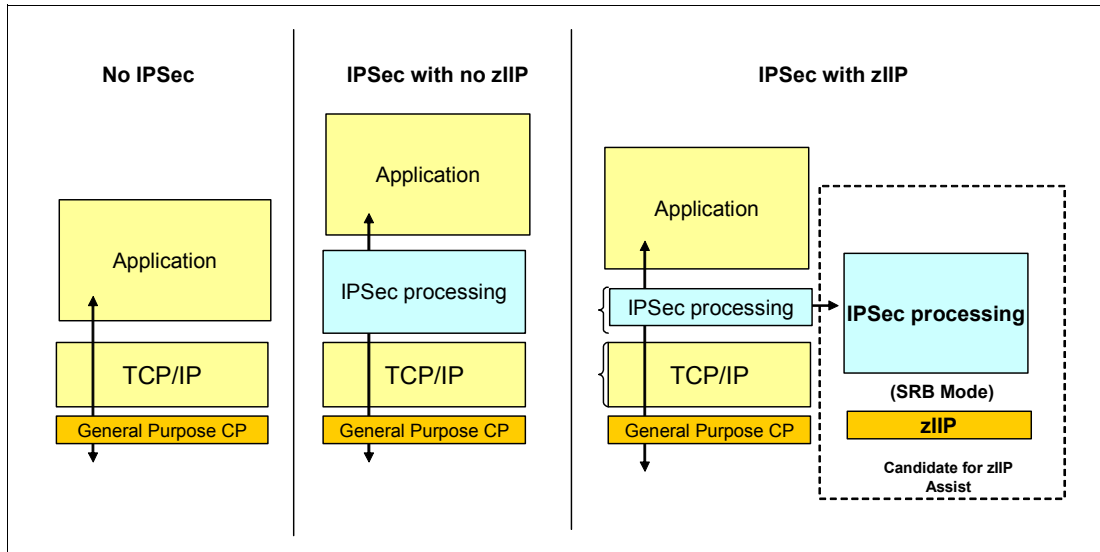


Figure 8-1 IPsec zIIP processing

AT-TLS is intended to protect traffic between specific client and server applications, and between the TCP/IP stacks to which these applications are bound. For application native SSL/TLS support, protection is only provided on data transported by regular TCP datagrams. The AT-TLS function only pertains to a z/OS endpoint. The other endpoint is required to support SSL or TLS (but can also be another z/OS endpoint using AT-TLS). It is also expected that SSL or TLS interoperability will not be a problem.

8.3 z/OS Encryption Facility

The Encryption Facility for z/OS (Program number: 5655-P97), first introduced in 2005, is a host-based software solution designed to encrypt sensitive data before transferring it to tape for archival purposes or business partner exchange. In addition to writing encrypted data to tape, the Encryption Facility for z/OS can also be used to produce encrypted data written to disk and other removable media. The Encryption Facility for z/OS (Encryption Facility) processes data at rest and is intended for encryption of media whose contents must be securely transported, that is, physically moved (for example, shipped in a truck, or electronically sent over non-secure links).

The security of the movement here covers both network eavesdropping and unauthorized reading of physical media containing sensitive information. Consequences of such an unauthorized disclosure of information can be severe, as illustrated by many examples in the press, where companies' finances and image were affected after losing track of physical media with sensitive contents known to be easily readable once one has access to the media itself.

IBM Encryption Facility for z/OS exploits the existing strengths of the mainframe and the IBM z/OS operating system. It is a host-based facility that uses existing centralized key management in z/OS and the hardware encryption capabilities of IBM mainframes.

When sharing encrypted data between z/OS systems, the Encryption Facility for z/OS software can also use the mainframe's hardware to compress the data, before the data is encrypted and written to the tape or disk media. A natural extension to the encryption of the

moved data is the encryption of archived data, thus the added encryption and decryption capabilities to the DFSMSdss DUMP and RESTORE functions as part of the features offered in the Encryption Facility for z/OS.

Encryption Facility for z/OS consists of two priced optional features:

- ▶ The Encryption Services feature supports encrypting and decrypting certain file formats on z/OS. This can allow you to transfer them to remote sites within your enterprise, transfer them to partners and vendors, and archive them. The Encryption Services feature supports both the System z format (introduced in Encryption Facility for z/OS V1.1) and the OpenPGP format (new with Encryption Facility for z/OS V1.2). The System z format supports hardware-accelerated compression before encryption.

The availability of the Encrypting Tape Drives, makes tapes more suitable option for encrypting data to be moved off-site while the Encryption Facility is a good option for sharing data between partners, with the OpenPGP support enhancing this option further. With the addition of the Encryption Facility for OpenPGP support in V1.2, you have two formats to choose from for handling your encryption needs when doing business partner data exchanges or for data exchanges within your own enterprise.

- ▶ The DFSMSdss Encryption feature enables the encryption of DFSMSdss dump data sets. This feature supports hardware-accelerated compression before encryption to tape.

Also available is the IBM Encryption Facility for z/OS Client. The Encryption Facility for z/OS Client is a no-cost, separately licensed program (which is offered as is, with no warranty) and is designed to enable the exchange of encrypted data between z/OS systems that have the Encryption Facility installed and systems running on z/OS and other platforms that needed the supported functions. Figure 8-2 summarizes the features.

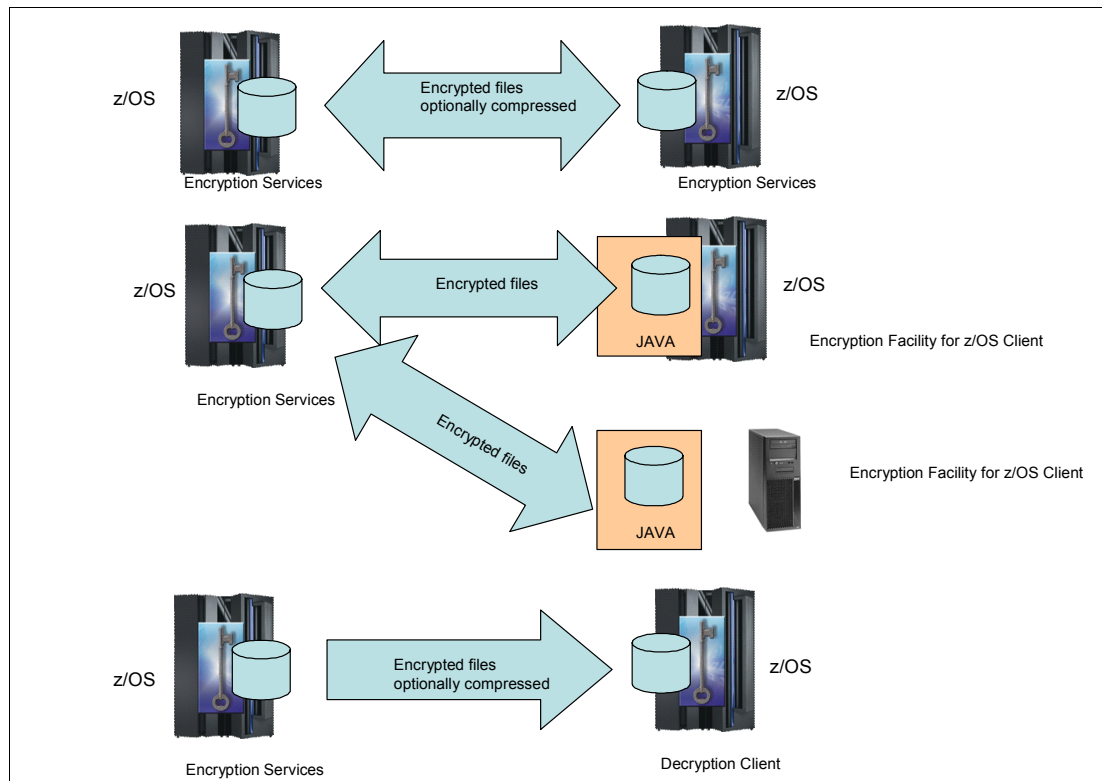


Figure 8-2 Encryption services and clients



Part 4

DB2 Audit Management Expert

DB2 Audit Management Expert is a tool that provides the auditing capabilities IT organizations need to minimize the liability associated with growing compliance demands.

In this part we provide details on the IBM Audit Management Expert tool.

- ▶ Chapter 9, “DB2 Audit Management Expert architecture and installation” on page 163
- ▶ Chapter 10, “Audit Management Expert scenarios” on page 211
- ▶ Chapter 11, “Audit Management Expert administration” on page 271



DB2 Audit Management Expert architecture and installation

In this chapter we describe the structure and the required steps to install DB2 Audit Management Expert.

This chapter contains the following sections:

- ▶ Architectural overview
- ▶ Storage modes
- ▶ Installation and configuration
- ▶ Security
- ▶ XML
- ▶ Data sharing
- ▶ Installing and configuring DB2 Audit Management Expert for z/OS

9.1 Architectural overview

DB2 Audit Management Expert for z/OS tracks, correlates, and audits activity using a proprietary SQL collector, the DB2 Instrumental Facility Interface (IFI), and a log analysis facility, and deposits audit information into a single repository to produce a complete view of this business activity for auditors.

There are several types of database events that can be tracked and audited. Some of these events include instances of denied access attempts without proper authorization, explicit grant and revoke statements, and the assignment and change of authorization IDs to access DB2. In addition, all DB2 commands and utilities can be audited, including monitoring selected objects for first changes of SQL insert, update, and delete statements, and recording the first read of audited objects using SQL Select and monitoring of create, alter, and drop operations.

DB2 Audit Management Expert uses audit data from the Audit SQL collector, the DB2 Trace Facility and log analysis. As it is captured, it is written to a single audit repository. A centralized repository creates the following advantages

- ▶ Consistency of views
- ▶ Single source for reporting which is available both online and in batch
- ▶ Institutional controls
- ▶ Summarization of the data
- ▶ High level trending of audit anomalies
- ▶ Drill-down capability -- a layer at a time
- ▶ Robust level of reporting events with minimal overhead controlled by the auditor without DBA involvement

As systems and data proliferate across the enterprise, centralization is integral to cost reduction, creating easier and more thorough audits, and reducing the risk of being out of compliance.

DB2 Audit Management Expert simplifies the process. It reduces manual auditing, takes large amounts of audit data and turns it into meaningful and manageable information and empowers non-technical users to easily audit the data without requiring log-ins to each system.

In a traditional environment, auditors require log-ins to all the Logical Partitions (LPARs) and require authorization to access each of the DB2 subsystems. In large sites, setting up and keeping track of all of these log-ins can be an administrative nightmare.

The DB2 Audit Management Expert administrator can specify how much visibility each auditor has to the auditable objects. Auditors are no longer required to issue DB2 commands to control the DB2 trace, as it is controlled within the product

Auditors using DB2 Audit Management Expert do not need to go to a large number of sources to get the data and they do not need user IDs for DB2 or the operating system on every system. They log into one place, DB2 Audit Management Expert, to have complete visibility of all auditable objects. An auditor can display collected data for all DB2 subsystems, or just the images and DB2 subsystems of interest, all from the central repository. The administration user interface, usually managed by the lead auditor, provides the ability to assign user IDs within the product to limit or allow auditor's access. For these reasons, DB2 Audit Management Expert makes auditing the data much more manageable.

Segregation of duties has always been a major challenge to the auditing process. In general, auditors usually depend on developers or database administrators (DBAs) to collect and report information. The most critical drawback of this approach pertains to the integrity of the audited information. Most database administrators have privileged user access. Privileged users need special authorities and access to DB2 resources to administer DB2 to perform their job. In the absence of a robust auditing mechanism to monitor the use of special privileges and data access patterns performed by privileged users, it is impossible to trace when or if these special privileges have been abused.

DB2 Audit Management Expert provides segregation of duties, resulting in data integrity, and accurate reports. This frees up DBAs to perform DBA duties and allows auditors to run audit reports independently of the DBAs, resulting in easier, more accurate audits. The continuous, automated auditing provided by DB2 Audit Management Expert removes the opportunity and the temptation to alter or omit, if authority permits, data from the audit reports. Thus, the independence of the audit mechanism from personal involvement provides assurance that data in the reports has not been modified, and consequently, the accuracy of data and reports is more reliable. Auditors now have the ability to adhere to published industry standards and external auditing without relying on personnel who need to be monitored.

The DB2 Audit Management Expert administrator can specify how much visibility each auditor has to the auditable objects. Auditors are no longer required to issue DB2 commands to control the DB2 trace, as it is controlled within the product.

9.1.1 General functions

Audit Management Expert collects and correlates data access information from a variety of DB2 resources to produce a comprehensive view of business activity for auditors:

- ▶ The proprietary Audit SQL collector (ASC) captures all SELECTS (reads) and all changes (UPDATE, INSERT, DELETE), dynamic and static SQL text, host variables and affected row counts for each SQL statement against DB2 tables.
- ▶ The IFI collector captures DDL: CREATE, ALTER(DB2 V9 only), DROP, authorization failures, grants and revokes, AUTHID changes, and command and utility executions in DB2 systems for DB2 tables.
- ▶ The log analysis facility captures before and after images for updates, after images for inserts, and before images for deletes to rows in an audited table (on demand).

The automated collection and correlation facility collects many different types of audit data as shown:

- ▶ All SELECTS (reads)
- ▶ All changes (UPDATE, INSERT, DELETE)
- ▶ CREATE, ALTER, DROP
- ▶ Explicit GRANT and REVOKE operations (to capture events where users may be attempting to modify authorization levels)
- ▶ IBM utility access
- ▶ Vendor utilities
- ▶ DB2 commands entered (including the ability to determine which users are issuing specific commands)
- ▶ Before and After images of changes (UPDATE), after images of INSERT, and before images of DELETE) to tables
- ▶ Dynamic and static SQL text for each statement and affected row counts
- ▶ Host variable value for each statement and affected row counts
- ▶ Assignment or modification of an authorization ID
- ▶ Authorization failures

The proprietary Audit SQL collector uses a collector developed in the IBM DB2 Query Monitor for z/OS. It is not necessary to have Query Monitor, but if you do, Query Monitor and the ASC component of DB2 Audit Management Expert for z/OS will use a shared master address space (shared collector) so that when both are running, the data is only collected once. If Query Monitor is not running, the DB2 Audit Management Expert ASC component starts the master address space. If the DB2 Audit Management Expert ASC component finds the master address space, it uses the master address space started by Query Monitor instead. If using both products, the overhead is significantly reduced.

A centralized repository is used to store audit data with the information auditors require about who performed what activity, and where, when, and how it was performed.

To collect the necessary data in the repository, a user needs to set up collection profiles, and collections. The administration user interface enables product administrators to define the data they want to be collected and activate those collections. It is essential to audit only data of interest, such as any data that is sensitive in nature and requires auditing. Filters are provided to help auditors collect the data in which they are interested. These are the activities that are truly useful to an auditor. If all data were collected, it would incur unnecessary overhead and require a huge repository. If it is a requirement to collect all data, the collected data can be filtered within the reports.

The reporting user interface provides flexible filters and other options for examining the data in the audit repository and to help auditors create meaningful reports.

The Log Analysis feature enables auditors to generate reports detailing the changes that have occurred in particular tables within a given time period. Log Analysis will display before and after images of changes (UPDATE), after images of INSERT, and before images of DELETE) to tables.

Many batch reports are available and can be automated through the system scheduler.

Support for multiple data storage modes to include Load repository, Generate off load data sets, and dual mode. See 9.2, "Storage modes" on page 168 for more information.

9.1.2 Components

Audit Management Expert is comprised of several components that work together to achieve processing.

Audit repository

The Audit Management Expert audit repository stores the audit data collected by Audit Management Expert.

Audit Management Expert captures the audit data and stores it in a set of DB2 tables referred to as the audit repository. The audit repository is DB2-based and contains tables to define DB2 systems and their properties. Each audit repository is associated with an Audit Management Expert server and is stand-alone (that is, repositories do not interact with other Audit Management Expert servers or repositories). Multiple audit repositories within the Audit Management Expert environment are permitted.

Server

The audit server is the central control point for a DB2 Audit Management Expert for z/OS. The server acts as the clearance for all user security (username/passwords), and starts the agent when a collection is activated. The server also stops the agent when a collection is changed to inactive.

The audit server provides services to support the following features:

- ▶ The administration user interface
- ▶ User administration
- ▶ To maintain collection criteria
- ▶ For audit repository administration
- ▶ The reporting user interface. Including accessing the audit repository on behalf of reporting users.
- ▶ Periodic summarization of collected audit data

The server runs as a batch job or started task on z/OS. A single audit server can support data collection from multiple agents auditing DB2 subsystems running on multiple z/OS systems

Agent

The DB2 Audit Management Expert for z/OS agent runs acts as a controller for the various collectors that are appropriate to the specific type of system on which the agent operates. The agent also maintains the necessary communications link to send information back and forth to the Audit Management Expert server. The agent runs as a batch job or started task on z/OS.

When using the Audit SQL Collector in conjunction with the IFI collector, a separate agent task will be necessary for each DB2 subsystem that will be audited using DB2 Audit Management Expert.

When using only the IFI Collector, one agent task will be necessary for each data sharing group or each stand alone DB2 subsystem being audited by DB2 Audit management Expert for z/OS.

The agent also performs normalization of collected audit data.

DB2 Instrumentation Facility Interface Collector

The IFI collector is contained within each agent task. A single agent can collect IFI data for all members of a data sharing group or an individual DB2 subsystem in a non-data sharing environment.

An Audit Management Expert server can have multiple agents distributed over a connected network-however, it is recommended that the Audit Management Expert server resides on the same LPAR as the Audit Repository.

Audit SQL Collector

The DB2 Audit Management Expert for z/OS Audit SQL collector (ASC) is responsible for collecting audit data related to SQL queries and selected other SQL statements performed against audited objects. This information is not available in the DB2 audit trace facility. This low overhead data collection provides for the capture of every SQL event for a set of audited objects, and capturing input host variable information and row count that SQL statement affects.

The audit SQL collector runs as a started task on a z/OS system. Each instance of the audit SQL collector can monitor one and only one DB2 subsystem. The audit SQL collector is started automatically by the DB2 Audit Management Expert for z/OS agent when requested.

User interfaces

Audit Management Expert provides two user interfaces:

- ▶ Administration (Audit Management Expert Administration)

The administration user interface enables Audit Management Expert product administrators to perform administrative tasks such as creating collections and collection profiles, and adding and modifying users.

- ▶ Reporting (Audit Management Expert Reporter)

The reporting user interface enables all Audit Management Expert users to generate and edit reports.

Log analysis

DB2 Audit Management Expert for z/OS provides an option to perform log analysis within the reporting user interface.

Log analysis for DB2 Audit Management Expert for z/OS allows authorized users to view who modified audited tables in a DB2 system and, if desired, to see the actual changes made. An interface (the Log Analysis tab within the DB2 Audit Management Expert for z/OS reporting user interface) is provided to enable you to input the information needed for log analysis. Selection capability is provided to allow you to refine the amount of information returned.

9.2 Storage modes

There are three storage modes available when using DB2 Audit Management Expert for z/OS. The available storage modes are described in this section.

Note: When DB2 Audit Management Expert for z/OS is used as the audit reporting mechanism, the Load Repository or Dual storage mode will be used. Dual or Generate off load data sets storage modes must be used when using supported third party vendor software.

9.2.1 Load repository mode

Load repository mode, shown in Figure 9-1 on page 169, is the default storage mode. Depending on the audit data requested, events are collected through the IFI collector, the Audit SQL Collector (ASC), or both. This mode loads all the audit data collected into the DB2 Audit Management Expert for z/OS repository tables.

DB2 Audit Management Expert Architecture Load Repository Mode

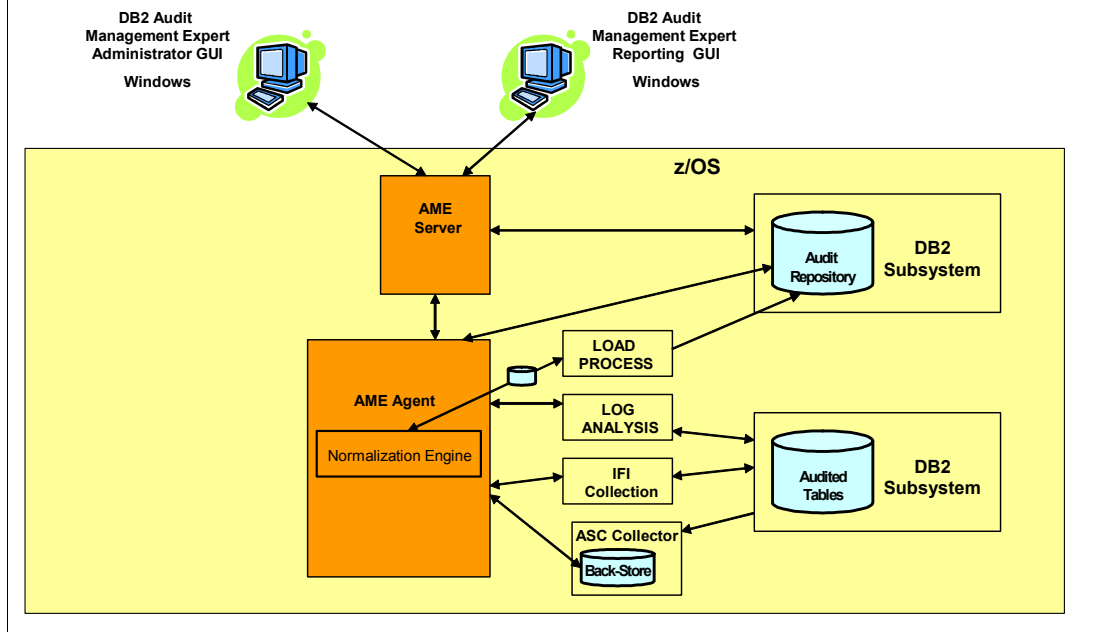


Figure 9-1 Load repository mode architecture

The agent is configured to load audit data into the DB2 Audit Management Expert for z/OS repository tables. The following conditions apply to this storage mode:

- ▶ When the audit SQL collector is used, the agent can be configured optionally to collect static SQL text.

If the agent has been configured to collect static SQL text, the text is periodically extracted from the DB2 catalog in the DB2 subsystem where the audit data was collected.

Note: Neither static or dynamic SQL text is collected unless the audit SQL collector is active.

- ▶ The INTERVAL command flushes all events that were collected up to the time that the INTERVAL command was issued.
 - Buffered events are flushed and loaded into the audit repository
 - The latest available data from the audit SQL collector is loaded into the audit repository.

Management of the data sets created by DB2 Audit Management Expert for z/OS is the responsibility of the agent.

9.2.2 Generate off load data sets mode

Generate off load data sets mode creates and populates off load data sets. The architecture of generate off load data sets mode is depicted in Figure 9-2 on page 170. Depending on the audit data requested, events are collected through the IFI collector, the audit SQL collector, or both.

Note: When using Generate Off load data sets mode, the Audit Repository is not updated.

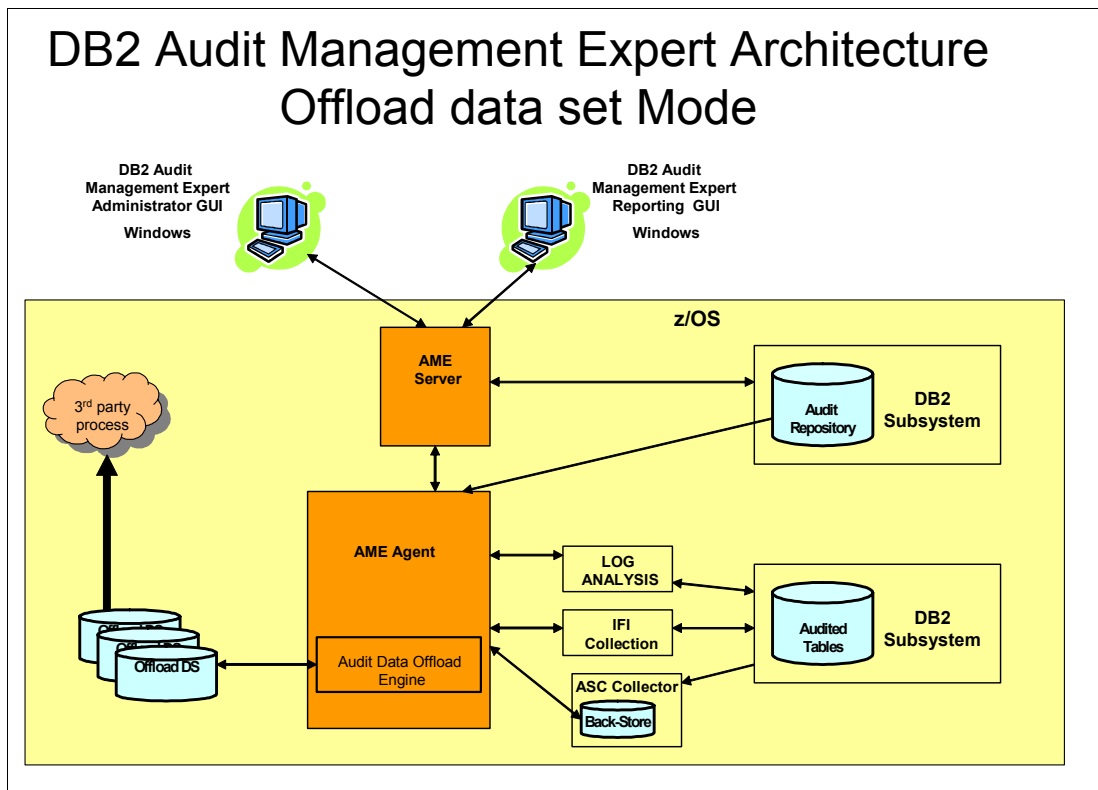


Figure 9-2 Generate Off load data set mode architecture

Data is written to off load data sets that are created by the DB2 Audit Management Expert for z/OS agent. The off load data sets are fully documented in the *IBM DB2 Audit Management Expert for z/OS User's Guide Version 2 Release 1, SC19-1302*. The off load data sets are then read and displayed by third party vendors or user-developed applications. Collected data is not written to the DB2 Audit Management Expert for z/OS audit repository.

The following conditions apply to this storage mode:

- ▶ When the audit SQL collector is used, the agent can be configured optionally to collect static SQL text.
- ▶ The collection of dynamic and static SQL text only occurs if ASC is activated.
- ▶ The INTERVAL command flushes all collected data to the off load data sets up to the time that the INTERVAL command was issued.
 - Initiated reading of the latest available audit SQL collector interval data sets
 - Closing of open off load data sets for the current interval
- ▶ The collected data is not normalized.

Note: The operation of off load data sets mode will cause additional resources to be consumed. The use of static SQL text collection mechanisms may result in a performance impact.

Management of the off load data sets is the responsibility of the user.

9.2.3 Dual mode

In the dual storage mode depicted in Figure 9-3, collected data is written to both the DB2 Audit Management Expert for z/OS audit repository and off load data sets. Depending on the audit data requested, events are collected through the IF I collector, the audit SQL collector, or both.

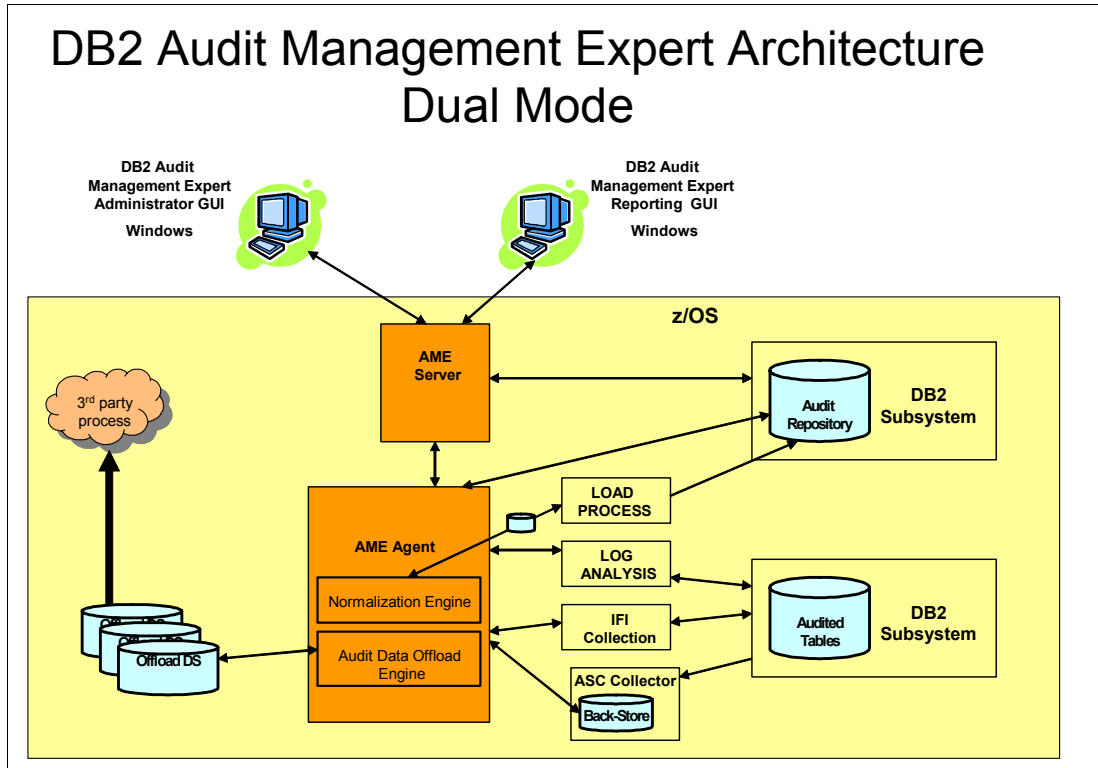


Figure 9-3 Dual mode architecture

The following conditions apply to this storage mode:

- ▶ When the audit SQL collector is used, the agent can be configured optionally to collect static SQL text.
- ▶ The collection of dynamic and static SQL text only occurs if audit SQL collector is activated.
- ▶ The INTERVAL command flushes all collected data to both the audit repository and the off load data sets up to the time that the INTERVAL command was issued.
 - Buffered events are flushed and loaded into the audit repository
 - The latest available data from the audit SQL collector is loaded into the audit repository.
 - Closing of open off load data sets for the current interval
- ▶ Data written to the off load data sets is not normalized.

Note: The operation of dual mode will cause additional resources to be consumed. The use of static SQL text collection mechanisms may result in a performance impact.

9.3 Installation and configuration

In this section we describe what to do for a new installation of DB2 Audit Management Expert for z/OS.

9.3.1 Planning for the installation

In this section, we examine the main prerequisites for the installation.

Software requirements

- ▶ Client
 - Windows XP Professional (32-bit).
 - DB2 Universal Database™ for Windows DB2 Connect (enables connection to the DB2 Audit Management Expert repository).
 - IBM DB2 JDBC Driver (V2.3.63 and higher)
 - To access a z/OS server using the JDBC Driver, the prerequisite JDBC package set must be bound on each server. Use the binder utility, DB2Binder, to bind the JDBC package set to your target database server

- ▶ Server and agent

Maintenance PK77147 (PTF 43941) is for z/OS V1R10 toleration. Verify its prerequisites.

SMP/E considerations

Detailed instructions for performing the SMP/E installation steps are included in *DB2 Audit Management Expert for z/OS Program Directory*, G110-8771-01. The following FMIDs are included with DB2 Audit Management Expert for z/OS:

- ▶ H35A210 DB2 Audit Management Expert
- ▶ H25F132 FEC Common Code

Note: FMID H25F132 contains common code and is shared among multiple IBM DB2 tools and is made available with multiple DB2 tools. The parent product for H25F132 is DB2 Change Accumulation Tool for z/OS, V01.04.00 (program number 5655-F55). When installing one of the tools that require the use of the FEC Common Code, it is highly recommended that FEC be brought up to the current maintenance level at the time of installation. If not, unpredictable results may occur.

APF library authorization

All libraries that are concatenated to the //STEPLIB DD statement must be APF authorized. For DB2@ Audit Management Expert for z/OS, these libraries end with the following suffixes:

- ▶ SADHLOAD
- ▶ SFECLOAD

Other libraries that may need to be APF authorized if they are not already include data sets are as follows:

- ▶ CEE.SCEERUN
- ▶ CEE.SCREERUN2
- ▶ SDNEXIT
- ▶ SDSNLOAD
- ▶ SYS1.LINKLIB

Started tasks and batch jobs

Both the servers and agents can be configured to run as started tasks or batch jobs. Other tasks can only run as started tasks. This is the time to decide how each of the DB2 Audit Management Expert for z/OS tasks will be executed.

Server

The number of servers should be considered before starting the installation of DB2 Audit Management Expert for z/OS. There is a minimum requirement of one server per sysplex. Depending on installation requirements, additional servers may be configured.

Important: Because each server has an audit repository unique to itself, the number of servers should be kept to a minimum.

The servers can run as either a batch job or a started task. The server configuration file is coded using XML syntax.

A sample server configuration file showing all available configuration parameters is included in Appendix B.1, “Server configuration file” on page 384. The server configuration file used for the server in this project is shown in Example 9-1.

Example 9-1 Server configuration file

```
<server-config>

  <server-repository>DB9A</server-repository>
  <client-listener-port>52522</client-listener-port>
  <agent-listener-port>52521</agent-listener-port>
  <object-qualifier>ADHTOOLS</object-qualifier>
  <summarizer-refresh-interval>300</summarizer-refresh-interval>
  <trace-network>false</trace-network>
  <trace-events>false</trace-events>
  <trace-config>true</trace-config>

</server-config>
```

Agent

There is one agent for each audited DB2 subsystem. An individual agent can connect to only one server. The agents can run as either a batch job or a started task. Agent programs should be given a high workload manager policy preference so they can keep up with the activity they are auditing.

Note: Implementing the agent as a started task will simplify the configuration.

The name of the agent task should include the SSID of the DB2 subsystem being audited by a given agent. This simplifies agent identification, administration, and maintenance.

Jobs submitted by the agent (through Log Analysis and DB2 Load processing) must be routed to a HELD output queue. If the DB2 Load or Log Analysis job is routed to a non-HELD job queue, the agent is unable to discern the status of the job.

A sample agent configuration file showing all available agent configuration parameters is included in Appendix B.2, “Agent configuration file” on page 389. The agent configuration file used for the agent in this project is shown in Example 9-2 on page 174.

Example 9-2 Agent configuration file

```
<agent-config>
  <server-address>wtsc63.itso.ibm.com</server-address>
  <agent-monitor>DB9A</agent-monitor>
  <server-repository>DB9A</server-repository>
  <object-qualifier>ADHTOOLS</object-qualifier>
  <object-collection>ADHCC210</object-collection>
  <server-port>52521</server-port>
  <log-level>I</log-level>
  <trace-csi>>false</trace-csi>
  <trace-ifi>>false</trace-ifi>
  <trace-network>>false</trace-network>
  <trace-db2-attachment>>false</trace-db2-attachment>
  <trace-sql>>false</trace-sql>
  <trace-db2loadstore>>false</trace-db2loadstore>
  <trace-db2load>>false</trace-db2load>
  <trace-norm>>false</trace-norm>
  <trace-events>>false</trace-events>
  <trace-config>>true</trace-config>
</agent-config>
```

Requirements for the Audit SQL Collector

The following requirements apply to the Audit SQL Collector.

Compatibility

Audit Management Expert Version 2.1 will not start auditing a DB2 subsystem that is running Query Monitor at Version 2.2 or lower level.

Note: Query Monitor and Audit SQL Collector are two separate functions and have no SMP/E installation dependencies. However, to inter-operate together within the same DB2 subsystem, Query Monitor must be above Version 2.2.

9.4 Security

The auditor must have confidence that the audit data in the repository accurately reflects the audited subsystems and has not been tampered with. Some guidelines are as follows:

- ▶ The server, the agents, and JDBC need different authorities. Use a separate user ID for each. The server and agent user IDs must have the ability to EXECUTE a SET CURRENT SQLID SQL statement using the Audit Repository schema name as the SQL ID. See *DB2 Version 9.1 for z/OS SQL Reference*, SC18-9854, for authorization necessary to execute this SQL statement. In addition to the authorization mentioned above, both the server and agent require a minimum of SYSCTRL authority to function properly.
 - Server: See SAMPLIB members ADHGRTS (repository tables), ADHGRTPS (packages) and ADHGRTQS (plans).
 - Agents: See SAMPLIB members ADHGRTA (repository tables) and ADHGRTQA (plans).
 - Reporting user interface (JDBC): See SAMPLIB member ADHGRTR (repository tables)(Give these user IDs no other authorities.)

- ▶ Create a collection profile that monitors repository table activity by any user ID other than these three.

Caution: Monitoring these three User IDs is recursive and will cause the repository to grow without limit.

- ▶ Segregation of duties: Ideally, production DBAs should not have access to the DB2 Audit Management Expert server and repository. This can be accomplished by creating a DB2 subsystem that contains only the Audit Repository and revoking the DBAs access after DB2 Audit Management Expert for z/OS has been installed and configured.
- ▶ The DB2 Audit Management Expert repository should be audited to ensure no one with authority has manipulated any audit data. Create a collection profile that monitors repository table activity by any user ID other than server and agent. A trusted context could be beneficial to further restrict the access to SYSCTRL for the server and the agent IDs.

Important: Auditing the server and agent User IDs is recursive and will cause the repository to grow without limit.

9.5 XML

Administrators can import or export collection profiles to other audit servers using XML files, or assign them to another DB2 system within the same audit server. The ability to import and export profiles allows for collection profiles that are created for common tables (such as SAP) to be created once and then used at other DB2 servers where the same application tables are installed.

Note: Details of XML syntax can be found in *DB2 Version 9.1 for z/OS XML Guide*, SC18-9858.

9.6 Data sharing

Audit Management Expert supports DB2 data sharing. Audit Management Expert uses DB2 data sharing to obtain audit information from all members of the data sharing group. You must run at least one instance of the DB2 Audit Management Expert for z/OS server to manage all of your DB2 subsystems and data sharing groups and to support all of your DB2 Audit Management Expert users. The server communicates with these clients and agents to perform Audit Management Expert functions.

9.7 Installing and configuring DB2 Audit Management Expert for z/OS

Table 9-1 shows the configuration steps and the corresponding SADHSAMP files.

Table 9-1 Configuration steps

Step and description	SADHSAMP member(s)
Step 1: APF authorizing the LOAD library data set	(Not applicable)
Step 2: Customizing DDL and JCL members using the ADHEMAC1 macro	ADHEMAC1
Step 3: Creating the Audit Management Expert control file	ADHSJ000
Step 4: Configuring the Audit Management Expert control file	ADHSJ001
Step 5: Configuring the audit repository	ADHDDLA, ADHDDLC, ADHDDL, ADHALSA, ADHALSS, ADHALSR, ADHGRTR1 ADHGRTA, ADHGRTR, ADHGRTS ADHBND70, ADHBND71,ADHBND72, ADHBND73, ADHBND74, ADHBND80, ADHBND81, ADHBND82, ADHBND83, ADHBND84, ADHBND90, ADHBND91, ADHBND92, ADHGRTB, ADHGRT7B, ADHGRT8B, ADHGRTPS, ADHGRTPR, ADHGRTQA, ADHGRTQS, and ADHGRTQR
Step 6: Configuring the server	ADHSJSRV,ADHCFG, ADHCFGSE, ADHBND73, ADHBND74, ADHBND72, ADHBND83, ADHBND84, ADHBND82, ADHBND92, ADHBND93, ADHBND94 and ADHGRTPS
Step 7: Configuring the Audit SQL Collector	ADHINTER, ADHCFGP, ADHCSSID
Step 8: Configuring the agent	ADHSJAGT, ADHCFGA,ADHCFGAE, ADHDDL, ADHBND72, ADHBND73,ADHBND74, ADHBND82,ADHBND83, ADHBND84,ADHBND92, ADHBND93,ADHBND94, ADHCFGP, ADHCFGPE, ADHGRT8A, and ADHGRT7A
Step 9: User administration procedure	ADHCFGU, ADHSUAP, and ADHSJUAP
Step 10: Installing the administration and reporting GUI clients	(not applicable)
Step 11: Deleting the password from the UAP configuration file	ADHCFGU
Step 12: Enabling audit data collection for reporting	ADHDDL
Step 13: Validating reporting access	(not applicable)

Table 9-2 shows the sample library members to install and configure DB2 Audit Management Expert.

Table 9-2 Installation and configuration sample library members

Description	Type	Member
Create aliases for the agent	JCL	ADHALSA
Customizable alias script for all parts of the product	JCL	ADHALSC
Drop aliases needed for the agent	JCL	ADHALSDA
Customizable drop alias script	JCL	ADHALSDC
Drop aliases needed for the reporting UI	JCL	ADHALSDR
Drop aliases needed for the server	JCL	ADHALSDS
Create aliases needed for the reporting UI	JCL	ADHALSR
Create aliases needed for the server	JCL	ADHALSS
Sample SQL statement for setting the AUDIT setting on for a table.	JCL	ADHALTT
DB2 V7 packages bind.	JCL	ADHBND70
DB2 V7 packages bind.	JCL	ADHBND71
DB2 V7 plans bind	JCL	ADHBND72
(For remote repository configuration only) Bind jobs required for V7.	JCL	ADHBND73
(For remote repository configuration only) Bind jobs required for V7.	JCL	ADHBND74
DB2 V8 packages bind.	JCL	ADHBND80
DB2 V8 packages bind.	JCL	ADHBND81
DB2 V8 plans bind.	JCL	ADHBND82
(For remote repository configuration only) Bind jobs required for V8.	JCL	ADHBND83
(For remote repository configuration only) Bind jobs required for V8.	JCL	ADHBND84
DB2 9 packages bind.	JCL	ADHBND90
DB2 9 packages bind.	JCL	ADHBND91
DB2 9 plans bind.	JCL	ADHBND92
(For remote repository configuration only) Bind jobs required for DB2 9.	JCL	ADHBND93
(For remote repository configuration only) Bind jobs required for DB2 9.	JCL	ADHBND94
Agent configuration file (contains the minimum number of configuration options for use in configuring an Audit Management Expert agent).	XML	ADHCFGGA
Agent configuration file (contains all of the possible options that can be used to configure an Audit Management Expert agent).	XML	ADHCFGAE
A minimal listing of parameters that control how the Audit SQL Collector is implemented.	80-byte sequential or PDS	ADHCFGGP
A full listing of parameters that control how the Audit SQL Collector is implemented.	80-byte sequential or PDS	ADHCFGPE

Description	Type	Member
Server configuration file (contains the minimum number of configuration options for use in configuring an Audit Management Expert server).	XML	ADHCFGGS
Server configuration file (contains all of the possible options that can be used to configure an Audit Management Expert server).	XML	ADHCFGSE
UAP configuration file (contains the minimum number of options for use in the update administrator password process).	XML	ADHCFGU
UAP configuration file (contains the all of the possible options for use in the update administrator password process).	XML	ADHCFGUE
Audit Management Expert ASC Started task procedure. Runs an instance of the Audit Management Expert ASC Started Task.	Procedure	ADHCSSID
DDL to create IBM DB2 SYSTOOLS database.	DDL	ADHDDLA
Creates the DB2 objects for Audit Management Expert.	JCL	ADHDDLCL
Drops the DB2 objects for Audit Management Expert.	DDL	ADHDDLDD
Drops aliases used with the Audit Management Expert server. Note: ADHDDLDS is used when only one user ID is to be used for all components (agent, server, report UI).	DDL	ADHDDLDS
Identifies the JDBC location	JCL	ADHDDL
JCL for creating optional indexes on repository event tables.	JCL	ADHDDLX1
(For remote repository configuration only) Creates the views on tables in the monitored DB2 needed by the agent.	JCL	ADHDDLRL
Creates aliases for use with the Audit Management Expert server. Note: ADHDDLSS is used when only one user ID is to be used for all components (agent, server, report UI).	JCL	ADHDDLSS
Creates aliases for IFI and ASC support	JCL	ADHDDLSS
Customizes the variables that appear in the DDL and JCL to be run.	(edit macro)	ADHEMAC1
Grants for the Audit Management Expert agent.	DDL	ADHGRTA
Grants privileges on the objects needed by DB2 Audit Management Expert (single user ID installation).	DDL	ADHGRTB
Grants privileges on the objects needed by DB2 Audit Management Expert (customizable scripts).	DDL	ADHGRTC
Grants to PLANS used by agent and server for DB2 Version 8.	DDL	ADHGRTPE8
Grants to PLANS used by agent and server for DB2 Version 9.	DDL	ADHGRTPE9
Grants for plans and packages for batch report ID.	JCL	ADHGRTPEPR
Grants for plans and packages for the server ID.	JCL	ADHGRTPEPS
Grants privileges for plans for the agent ID	JCL	ADHGRTPEQA
Grants privileges for plans for reporting ID (batch reports).	JCL	ADHGRTPEQR
Grants privileges for plans for the server ID.	JCL	ADHGRTPEQS
Grants for the Audit Management Expert reporting interface.	DDL	ADHGRTPETR
Grants for the Audit Management Expert server.	DDL	ADHGRTPESTS

Description	Type	Member
Plan and package grants.	SQL	ADHGRT1
Grants for plans and packages for the agent ID (DB2 V7).	JCL	ADHGRT7A
Grants for plans and packages for one ID (DB2 V7).	JCL	ADHGRT7B
Grants for plans and packages for the agent ID (DB2 V8).	JCL	ADHGRT8A
Grants for plans and packages for one ID (DB2 V8).	JCL	ADHGRT8B
Grants for plans and packages for the agent ID (DB2 9).	JCL	ADHGRT9A
Grants for plans and packages for one ID (DB2 9).	JCL	ADHGRT9B
JCL to create the ASC VSAM Interval Datasets.	JCL	ADHINTER
Produces various batch reports from the Audit Management Expert repository.	JCL	ADHRPRT
(For submitted jobs) Runs an instance of the Audit Management Expert agent to audit a single DB2 subsystem on one LPAR.	JCL	ADHSJAGT
(For submitted jobs) Runs an instance of the Audit Management Expert agent server to audit a single DB2 subsystem on one LPAR.	JCL	ADHSJSRV
JCL to submit to invoke the module ADHSUAP.	JCL	ADHSJUAP
Allocate VSAM product control file.	JCL	ADHSJ000
Set product configuration options.	JCL	ADHSJ001
Product control file content report generator.	JCL	ADHSJ003
(For started tasks) Runs an instance of the Audit Management Expert agent to audit a single DB2 subsystem on one LPAR.	Procedure	ADHSPAGT
(For started tasks) Runs an instance of the Audit Management Expert server to audit a single DB2 subsystem on one LPAR.	Procedure	ADHSPSRV

Step 1: APF authorizing the LOAD library data set

DB2 Audit Management Expert requires that the following library is APF authorized.

- ▶ adhh1q.SADHLOAD
- ▶ fechlq.SFECLOAD
- ▶ CEE.SCEERUN
- ▶ CEE.SCEERUN2
- ▶ DB2 EXIT
- ▶ DSN.V9R1.SDSNLOAD

Step 2: Customizing DDL and JCL using the ADHEMAC1 macro

The edit macro ADHEMAC1 provides a straightforward way to customize the variables that appear in the DDL and JCL that will be run. Copy member ADHEMAC1 from the adhh1vl.SADHSAMP to your site's CLIST library and edit the ADHEMAC1 macro with the appropriate variables. We run ADHEMAC1 macro to update member variables, typing ADHEMAC1 when we are edit a member. See Example 9-3 on page 180.

Tip: Create a copy of the ADHEMAC1 edit macro for each DB2 subsystem that will be audited. This simplifies the configuration of DB2 Audit Management Expert for z/OS for each audited DB2 subsystem.

Example 9-3 ADHEMAC1 macro customized

```
/* Member: ADHEMAC1 */
/* */
/* 5655-T57 */
/* Copyright IBM Corp. 2004, 2008 All Rights Reserved. */
/* Copyright Rocket Software, Inc. 2004 - 2008 All Rights */
/* Reserved. */
/* */
/* */
/* Product installation assist ISPF edit macro. */
/* */
/* */
ISREDIT MACRO (NP)
ISPEXEC VGET (ZUSER)
ISREDIT CHANGE ALL '#SSID' DB9A
ISREDIT CHANGE ALL '#ADHOWNER' &ZUSER
ISREDIT CHANGE ALL '#ADHQUALIFIER' ADHTOOLS
ISREDIT CHANGE ALL '#ADHDATABASE' ADHTOOLS
ISREDIT CHANGE ALL '#ADHUSERID' &ZUSER
ISREDIT CHANGE ALL '#ADHAGENTID' PAOLOR1
ISREDIT CHANGE ALL '#ADHSERVERID' PAOLOR1
ISREDIT CHANGE ALL '#ADHREPORTID' PAOLOR1
ISREDIT CHANGE ALL '#STGX' ADHSG01
ISREDIT CHANGE ALL '#VCAT' DB9AU
ISREDIT CHANGE ALL '#SADHLOAD' ADH.V2R1M0.SADHLOAD
ISREDIT CHANGE ALL '#SADHMENU' ADH.V2R1M0.SADHMENU
ISREDIT CHANGE ALL '#SADHSLIB' ADH.V2R1M0.SADHSLIB
ISREDIT CHANGE ALL '#SADHDBRM' ADH.V2R1M0.SADHDBRM
ISREDIT CHANGE ALL '#SDSNLOAD' DB9A9.SDSNLOAD
ISREDIT CHANGE ALL '#SDSNRUNL' DB9AU.RUNLIB.LOAD
ISREDIT CHANGE ALL '#DSNTEP2' DSNTEP2
ISREDIT CHANGE ALL 'ADHPLAN1' ADHPLAN1
ISREDIT CHANGE ALL 'ADHPLAN2' ADHPLAN2
ISREDIT CHANGE ALL 'ADHPLAN3' ADHPLAN3
ISREDIT CHANGE ALL 'ADHPLAN4' ADHPLAN4
ISREDIT CHANGE ALL '#SZPARM' DSNZPARM
ISREDIT CHANGE ALL '#SBSDS01' DB9AU.BSDS01
ISREDIT CHANGE ALL '#SBSDS02' DB9AU.BSDS02
ISREDIT CHANGE ALL '#SDSNEXIT' DB9A9.SDSNEXIT
ISREDIT CHANGE ALL '#ADHASCINHLQ' ADH.SC63.INTERVAL
ISREDIT CHANGE ALL '#ADHASCDDLQ' ADHUSER.DB9A
ISREDIT CHANGE ALL '#ADHASCSC'
ISREDIT CHANGE ALL '#SFECLOAD' FEC.V2R1M0.SFECLOAD
ISREDIT CHANGE ALL '#ADHCUSTOMID' CUSTID
ISREDIT CHANGE ALL '#ADHCUSTOMID1' CUSTID1
ISREDIT CHANGE ALL '#ADHCUSTOMID2' CUSTID2
ISREDIT CHANGE ALL '#ADHCUSTOMID3' CUSTID3
ISREDIT CHANGE ALL '#ADHCUSTOMID4' CUSTID4
ISREDIT CHANGE ALL '#ADHTMPPDS' ADH.TMP
ISREDIT CHANGE ALL '#ADHCNTRLFILE' ADH.V2R1M0.CONTROL
```

Step 3: Creating the Audit Management Expert control file

IBM DB2 Audit Management Expert configuration information is stored in a VSAM data set referred to as the product control file. See Example 9-4 on page 181.

Note: Control files can be shared between subsystem, but if the IBM DB2 Log Analysis Tool is in use, you must use a separate VSAM control file for IBM DB2 Audit Management Expert.

Example 9-4 ADHSJ000: Create Audit Management Expert control file

```
//IDCAMS EXEC PGM=IDCAMS
//*****
/* 5655-T57
/* Copyright Rocket Software, Inc. 2004 - 2008 All Rights
/* Reserved.
//*****
/* IBM DB2 Audit Management Expert for z/OS 2.1.0
/*
/* THIS JCL WILL CREATE THE PRODUCT VSAM CONTROL DATA SET, ALSO
/* REFERRED TO AS THE DB2 CONTROL DATA SET.
/*
/* CAUTION: THIS IS NEITHER A JCL PROCEDURE NOR A COMPLETE JOB.
/*
/* Before running this job, you will have to make the following
/* modifications:
/*
/* 1. Change the job statement to meet your system requirements
/*
/* 2. Change ADH.V2R1M0.CONTROL below to the name of the VSAM
/* control data set to be created.
/*
/* 3. Change/remove the VOLUMES parameter.
/*
//*****
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSUT1 DD *
DUMMY RECORD TO SATISFY VSAMS REQUIREMENT FOR A NON-EMPTY INITIAL FILE.
/*
//SYSIN DD *
DELETE 'ADH.V2R1M0.CONTROL'

SET MAXCC = 0

DEFINE CLUSTER                                -
  ( NAME ('ADH.V2R1M0.CONTROL')              -
    VOLUMES (SBOXB4)                          -
    CYLINDERS (1 1)                            -
    SHAREOPTIONS (2 3)                         -
    INDEXED                                    -
    RECORDSIZE (1024 1024)                     -
    KEYS (32 0)                                -
    REUSE )                                    -
  DATA ( NAME ('ADH.V2R1M0.CONTROL.DATA')) -
  INDEX ( NAME ('ADH.V2R1M0.CONTROL.INDEX'))

REPRO INFILE(SYSUT1) OUTDATASET('ADH.V2R1M0.CONTROL') REUSE

/*
//
```

Step 4: Configuring the Audit Management Expert control file

Audit Management Expert requires information that identifies target DB2 subsystems, product execution options, and data set attributes. The product configuration is saved in the VSAM product control file data set that you created in the previous step. See Example 9-5.

Note: In a data sharing environment, specify subsystem names (not groupnames).

Example 9-5 ADHSJ001: Configure Audit Management Expert control file

```
//*****
//*
//* 5655-T57
//* Copyright IBM Corp. 2004, 2008 All Rights Reserved.
//* Copyright Rocket Software, Inc. 2004 - 2008 All Rights
//* Reserved.
//*
//*****
//* IBM DB2 Audit Management Expert for z/OS 2.1.0
//*
//*-----*
//*
//*PCFUPDT EXEC PGM=ADH#UTIL,PARM='SETUP',REGION=4M
//*STEPLIB DD DISP=SHR,DSN=ADH.V2R1M0.SADHLOAD
//*SYSUDUMP DD SYSOUT=*
//*DB2PARMS DD DISP=SHR,DSN=ADH.V2R1M0.CONTROL
//*
//* REPORTS
//*
//*SYSOUT DD SYSOUT=*,RECFM=FBA,LRECL=133 SYSIN REPORT
//*SYSPRINT DD SYSOUT=*,RECFM=FBA,LRECL=133 PCF REPORT
//*
//* CONTROLS
//*
//*SYSIN DD *
*
*-----*
* Sample statements to add/update DB2 subsystem information.
* Multiple sets of following DB2 information control statements
* can be created and run in a single setup run.
*-----*
*
SET DB2 SSID = DB9A
UPDATE DB2 ZPARMS = DSNZPARM
UPDATE DB2 BOOTSTRAP1 = DB9AU.BSDS01
UPDATE DB2 BOOTSTRAP2 = DB9AU.BSDS02
UPDATE DB2 LOADLIB1 = DB9A9.SDSNEXIT
UPDATE DB2 LOADLIB2 = DB9A9.SDSNLOAD
*UPDATE DB2 LOADLIB3 =
*UPDATE DB2 LOADLIB4 =
*UPDATE DB2 LOADLIB5 =
*
*-----*
* Sample statements to add/update ADH product plans
*-----*
*
SET DB2 SSID = DB9A
SET PRODUCT CFG = NULL
SET PRODUCT VER = NULL
*
```

```

UPDATE ADH PLAN1      = ADHPLAN1  AUDIT MANAGEMENT EXPERT SERVICES
UPDATE ADH PLAN2      = ADHPLAN2  LOG ANALYSIS SERVICES
UPDATE ADH PLAN3      = ADHPLAN3  AUDIT MANAGEMENT EXPERT SERVICES
UPDATE ADH PLAN4      = ADHPLAN4  ASC SERVICES
UPDATE ADH CORR ID 1  = ADH ID 1
UPDATE ADH CORR ID 2  = ADH ID 2
UPDATE ADH COLLECTION =
UPDATE ADH ACCT TOKEN = ACCOUNTING ADH
UPDATE ADH SCOPE       = SCOPE(GROUP)
UPDATE ADH IFI INFO    = CLASS(1,2,3,4,5,7,8) IFCID(90,91)
UPDATE ADH MSGLIBRARY = ADH.V2R1M0.SADHMENU
UPDATE ADH ARCHLOG1   = N          USE ARCHIVE LOG 1
UPDATE ADH ARCHLOG2   = N          USE ARCHIVE LOG 2
UPDATE ADH ACTLOGPRI  = Y          ACTIVE LOG PRIORITY
*
*-----
* Sample statements to add/update default z/OS JCL JOB statements
*-----
*
UPDATE ADH JOB STMTS:
//ADHJOB  JOB STATEMENT
//*
<END>
*-----
* Sample statements to add/update log analysis services ROWDATA
* VSAM data set attributes.
*-----
*
* The ROWDATA VSAM data set is dynamically created by the log
* analysis services. The VSAM data set name by default is prefixed
* with user id. The VSAM data set prefix can be altered using the
* DSN PFX control listed below. If volumes need to be specified when
* the VSAM data set is created use the VOLS statement listed below
* to supply a maximum of 3 volsers.
*
*UPDATE LAS VSAM DSN PFX= XXXXXXXXXXXXXXXXXXXX (Max length 21)
UPDATE LAS VSAM VOLS  = SBOXED,SBOXEF (1-3 volsers)
*
*-----
* LAS temporary 'work' file HLQ PREFIX
*-----
*
* During Log Analysis processing, the LAS component may need to
* create temporary 'work' files. The default prefix for the location of
* these files is the user id concatenated with the string '.ADHLAT'.
* This location can be altered using the DATASET PFX listed below.
*
*UPDATE LAS DATASET PFX = XXXXXXXXXXXXXXXXXXXX (Max length 17)
*
* Product output Unicode data conversion information.
* These should not be altered.
*
UPDATE CCS ADH TECHNQ = ER          CHARACTER CONVERSION TECHNIQUE
UPDATE CCS ADH SBCS  = 00037       EBCDIC SBCS CCSID
UPDATE CCS ADH DBCS  = 01200       UNICOD E UT-8 DBCS CCSID
UPDATE CCS ADH MIXED = 01208       UNICOD E UT-8 MIXED CCSID
*
*-----
* Sample statements to add/update default data set information
*-----

```

```

*
* File tailoring work data set allocation.
*
UPDATE FTW DEVICE      = SYSALLDA      DEVICE TYPE
UPDATE FTW ALCUNIT    = C              C=CYLS, T=TRACKS
UPDATE FTW PQTY       = 00001         PRIMARY QTY
UPDATE FTW SQTY       = 00001         SECONDARY QTY
*UPDATE FTW SMSDC     = xxxxxxxx      SMS DATA CLASS
*UPDATE FTW SMSSC     = xxxxxxxx      SMS STORAGE CLASS
*UPDATE FTW SMSMC     = xxxxxxxx      SMS MANAGEMENT CLASS
*
/*
//

```

Step 5: Configuring the audit repository

Follow the six sub-steps (A to F) to configure Audit Management Expert audit repository.

A- Creating Audit Management Expert DB2 objects

We use ADHEMAC1 edit macro to edit the DDL.

- ▶ Edit and run the ADHDDLA member in the adhhilvl.SADHSAMP library to create database.
- ▶ Modify and run member ADHDDLC to create Audit Management Expert repository tables and views.
- ▶ Modify and run member ADHDDL5 to create the aliases for the audit repository.

Note: This member must be run once for the schema name used for the Audit Repository. The job must also be run once for each user ID assigned to the server, agent, and reporting client.

See Example 9-6.

Example 9-6 ADHDDLA: Create Audit Management Expert audit repository

```

//*****
/* IBM DB2 AUDIT MANAGEMENT EXPERT FOR Z/OS 2.1.0
/*
/* 5655-T57
/* COPYRIGHT IBM CORP. 2003, 2008 ALL RIGHTS RESERVED.
/* COPYRIGHT ROCKET SOFTWARE, INC. 2003 - 2008 ALL RIGHTS
/* RESERVED.
/*
/* USE THIS JCL FOR DB2 VERSIONS 7, 8 AND 9.
/*
/* THIS MEMBER PROVIDES THE DDL THAT CAN BE USED TO CREATE THE
/* ADHTOOLS DATABASE. SAMPLE DDL TO CREATE A ADHTOOLS
/* STOGROUP IS ALSO PROVIDED.
/*
//*****
//ADHDDLA EXEC PGM=IKJEFT01,COND=(4,LT),DYNAMNBR=20
//STEPLIB DD DISP=SHR,DSN=DB9A9.SDSNLOAD
/*
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTEM DD SYSOUT=*
//CEEDUMP DD SYSOUT=*

```



```

//SYSUDUMP DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSTSIN DD *
DSN SYSTEM(DB9A)
RUN PROGRAM(DSNTEP2) PLAN(DSNTEP2) -
      LIB('DB9AU.RUNLIB.LOAD') PARS(' /ALIGN(MID) ')
END
//SYSIN DD *
--
-- CREATE ADHTOOLS STORAGE GROUP
--
-- THE DDL TO CREATE A STORAGE GROUP IS PROVIDED AS A SAMPLE.
--
DROP STOGROUP ADHSG01;

CREATE STOGROUP ADHSG01
      VOLUMES('*')
      VCAT DB9AU;
COMMIT;

--
-- CREATE ADHTOOLS DATABASE
--
-- IF CREATING THE ADHTOOLS DATABASE ON A DB2 V8 NFM SUBSYSTEM
-- OR DB2 V9 (ANY TYPE), CHANGE THE BUFFER POOL VALUE TO BP8K0.
--
DROP DATABASE ADHTOOLS;

CREATE DATABASE ADHTOOLS
      BUFFERPOOL BP1
      STOGROUP ADHSG01
      CCSID UNICODE;
COMMIT;

/*
/**

```

The detailed configuration file for the DDL associated with the Audit Management Expert repository objects is available from the library ADH.V2R1M0.SADHSAMP.

Here we list some of the customization jobs. See Example 9-7.

Example 9-7 ADHDDL: Create Alias

```

//PAOLR16 JOB (XXX,POK),DRJOB2,CLASS=A,MSGCLASS=X
/*JOBPARM SYSAFF=SC63
/**
*****
/**
/** MEMBER: ADHDDL
/**
/**-----*
//JOBLIB DD DISP=SHR,DSN=DB9A9.SDSNLOAD
//DSNTIAS EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
      DSN SYSTEM(DB9A)

```

```

RUN PROGRAM(DSNTEP2) PLAN(DSNTEP2) -
LIB('DB9AU.RUNLIB.LOAD')
END
/*
//SYSIN DD *
-- START OF Z/OS STATEMENTS *****
--
-- SET CURRENT SQLID = 'SYSADM';
--

CREATE ALIAS ADHTOOLS.ADHSUSER_MASTER
FOR ADHTOOLS.ADHVUSER_MASTER;
CREATE ALIAS ADHTOOLS.ADHSPERMISSION
FOR ADHTOOLS.ADHVPERMISSION;
CREATE ALIAS ADHTOOLS.ADHSCOL_PRFL
FOR ADHTOOLS.ADHVCOL_PRFL;
CREATE ALIAS ADHTOOLS.ADHSAGENT
FOR ADHTOOLS.ADHVAGENT;
CREATE ALIAS ADHTOOLS.ADHSCOLLECTION
FOR ADHTOOLS.ADHVCOLLECTION;
CREATE ALIAS ADHTOOLS.ADHSXGROUP
FOR ADHTOOLS.ADHVXGROUP;
CREATE ALIAS ADHTOOLS.ADHSUSER_GROUP_MAP
FOR ADHTOOLS.ADHVUSER_GROUP_MAP;
CREATE ALIAS ADHTOOLS.ADHSGROUP_PERM_MAP
FOR ADHTOOLS.ADHVGROUP_PERM_MAP;
CREATE ALIAS ADHTOOLS.ADHSUSER_PERM_MAP
FOR ADHTOOLS.ADHVUSER_PERM_MAP;
CREATE ALIAS ADHTOOLS.ADHSENTITY
FOR ADHTOOLS.ADHVENTITY;
CREATE ALIAS ADHTOOLS.ADHSENTITY_TYPE
FOR ADHTOOLS.ADHVENTITY_TYPE;
CREATE ALIAS ADHTOOLS.ADHSEVENT
FOR ADHTOOLS.ADHVEVENT;
CREATE ALIAS ADHTOOLS.ADHSEVENT_ACTION
FOR ADHTOOLS.ADHVEVENT_ACTION;
CREATE ALIAS ADHTOOLS.ADHSEVENT_TARGET
FOR ADHTOOLS.ADHVEVENT_TARGET;
CREATE ALIAS ADHTOOLS.ADHSEVENT_USER
FOR ADHTOOLS.ADHVEVENT_USER;
CREATE ALIAS ADHTOOLS.ADHSEVNT_SRC_DTAIL
FOR ADHTOOLS.ADHVEVNT_SRC_DTAIL;
CREATE ALIAS ADHTOOLS.ADHSEVENT_SRC
FOR ADHTOOLS.ADHVEVENT_SRC;
CREATE ALIAS ADHTOOLS.ADHSEVENT_DTAIL
FOR ADHTOOLS.ADHVEVENT_DTAIL;
CREATE ALIAS ADHTOOLS.ADHSEVENT_AUTCHNG
FOR ADHTOOLS.ADHVEVENT_AUTCHNG;
CREATE ALIAS ADHTOOLS.ADHSEVENT_TEXT
FOR ADHTOOLS.ADHVEVENT_TEXT;
CREATE ALIAS ADHTOOLS.ADHSEVENT_EXEC
FOR ADHTOOLS.ADHVEVENT_EXEC;
CREATE ALIAS ADHTOOLS.ADHSAUTH_PRFL
FOR ADHTOOLS.ADHVAUTH_PRFL;
CREATE ALIAS ADHTOOLS.ADHSREPORT_PRFL
FOR ADHTOOLS.ADHVREPORT_PRFL;
CREATE ALIAS ADHTOOLS.ADHSRFL
FOR ADHTOOLS.ADHVPRFL;
CREATE ALIAS ADHTOOLS.ADHSCOL_PRFL_RL_MP
FOR ADHTOOLS.ADHVCOL_PRFL_RL_MP;

```

```

CREATE ALIAS ADHTOOLS.ADHSRULE_EVENT
FOR ADHTOOLS.ADHVRULE_EVENT;
CREATE ALIAS ADHTOOLS.ADHSRULE_SCHEDULE
FOR ADHTOOLS.ADHVRULE_SCHEDULE;
CREATE ALIAS ADHTOOLS.ADHSRULE_TARGET
FOR ADHTOOLS.ADHVRULE_TARGET;
CREATE ALIAS ADHTOOLS.ADHSRULE_USER
FOR ADHTOOLS.ADHVRULE_USER;
CREATE ALIAS ADHTOOLS.ADHSRULE_APP
FOR ADHTOOLS.ADHVRULE_APP;
CREATE ALIAS ADHTOOLS.ADHSRULE_PLAN
FOR ADHTOOLS.ADHVRULE_PLAN;
CREATE ALIAS ADHTOOLS.ADHSRULE
FOR ADHTOOLS.ADHVRULE;
CREATE ALIAS ADHTOOLS.ADHSDATABASE
FOR ADHTOOLS.ADHVDATABASE;
CREATE ALIAS ADHTOOLS.ADHSJOB
FOR ADHTOOLS.ADHVJOB;
CREATE ALIAS ADHTOOLS.ADHSAUDIT_MAP
FOR ADHTOOLS.ADHVAUDIT_MAP;
CREATE ALIAS ADHTOOLS.AHSDUMMY1
FOR ADHTOOLS.ADHVDUMMY1;
CREATE ALIAS ADHTOOLS.AHSCONTEXT
FOR ADHTOOLS.ADHVCONTEXT;
CREATE ALIAS ADHTOOLS.ADHS_METRICSINFO
FOR ADHTOOLS.ADHV_METRICSINFO;
CREATE ALIAS ADHTOOLS.ADHS_SUMMARYUPDATE
FOR ADHTOOLS.ADHV_SUMMARYUPDATE;
CREATE ALIAS ADHTOOLS.AHSSUMMARYUSER
FOR ADHTOOLS.ADHVSUMMARYUSER;
CREATE ALIAS ADHTOOLS.ADHS_TL_HOUSER
FOR ADHTOOLS.ADHV_TL_HOUSER;
CREATE ALIAS ADHTOOLS.ADHS_TL_DAYUSER
FOR ADHTOOLS.ADHV_TL_DAYUSER ;
CREATE ALIAS ADHTOOLS.ADHS_TL_WEEKUSER
FOR ADHTOOLS.ADHV_TL_WEEKUSER ;
CREATE ALIAS ADHTOOLS.ADHS_TL_MONTHUSER
FOR ADHTOOLS.ADHV_TL_MONTHUSER;
CREATE ALIAS ADHTOOLS.ADHS_DBNAME
FOR ADHTOOLS.ADHV_DBNAME;
CREATE ALIAS ADHTOOLS.ADHS_TEXT_XREF
FOR ADHTOOLS.ADHV_TEXT_XREF;
CREATE ALIAS ADHTOOLS.ADHS_CLLCTR_TYPE
FOR ADHTOOLS.ADHV_CLLCTR_TYPE;
CREATE ALIAS ADHTOOLS.ADHS_ASCCHPT
FOR ADHTOOLS.ADHV_ASCCHPT;
CREATE ALIAS ADHTOOLS.ADHS_DLCHPT
FOR ADHTOOLS.ADHV_DLCHPT;
CREATE ALIAS ADHTOOLS.ADHS_METADATA
FOR ADHTOOLS.ADHV_METADATA;
CREATE ALIAS ADHTOOLS.ADHSEVENT_HOSTVS
FOR ADHTOOLS.ADHVEVENT_HOSTVS;
CREATE ALIAS ADHTOOLS.ADHS_FILTERS
FOR ADHTOOLS.ADHV_FILTERS;
-- NOTE: THE ALIAS FOR THE ADH_V7SEQ TABLE MUST POINT AT A
-- 'REAL' TABLE AND NOT A VIEW.
CREATE ALIAS ADHTOOLS.ADHS_V7SEQ
FOR ADHTOOLS.ADH_V7SEQ;
CREATE ALIAS ADHTOOLS.ADHS_DATATYPE
FOR ADHTOOLS.ADHV_DATATYPE;

```

```

CREATE ALIAS ADHTOOLS.ADHS_STSQLUPDATE
FOR ADHTOOLS.ADHV_STSQLUPDATE;

COMMIT;
/*
//

```

B- Granting Audit Management Expert privileges

This section describes how to grant privileges that are required to access Audit Management Expert. The installation user ID should differ from the agent, server, or reporting user ID that is used to run the grant jobs.

Notes: Under most conditions, you can control product access at the product plan level.

You do not need to run ADHGRTA if the user ID used for installation purposes will also run the agent.

You do not need to run ADHGRTS if the user ID used for installation will also run the server.

You do not need to run ADHGRTR if the user ID used for installation will also be used for reporting.

- ▶ Run member ADHGRTB to grant the installer user ID access to objects.
- ▶ Run member ADHGRTA to authorize the DB2 authorization ID that will run the Audit Management Expert agent.
- ▶ Run member ADHGRTS to authorize the DB2 authorization ID that will run the Audit Management Expert server.
- ▶ Run member ADHGRTR to authorize the DB2 authorization ID to be used for reporting.

C- Binding Audit Management Expert DB2 packages

Bind the Audit Management Expert packages for each subsystem against which you plan to run DB2 Audit Management Expert. We installed Audit Management Expert on DB2 9 for local connection, so we execute jobs ADHBND90 and ADHBND91 only.

- ▶ Installing in DB2 9:
 - For local connection
 - ADHBND90—Bind Audit Management Expert packages for the repository. See Example 9-8.

Example 9-8 ADHBND90: Bind Audit Management Expert packages for the repository.

```

//*****
/* Member: ADHBND90
/* 5655-T57
/* Copyright Rocket Software, Inc. 2004 - 2008 All Rights
/* Reserved.
/*-----*
/*
/* Bind Step - Bind Product Packages
/*
//BIND1      EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB   DD DISP=SHR,DSN=DB9A9.SDSNLOAD
//SYSPRINT  DD SYSOUT=*
//SYSTSPRT  DD SYSOUT=*

```

```

//SYSUDUMP DD SYSOUT=*
//DBRMLIB DD DISP=SHR,DSN=ADH.V2R1MO.SADHDBRM
//SYSTSIN DD *
    DSN SYSTEM(DB9A)
*
* ADH PRODUCT PACKAGES GO HERE
*
BIND PACKAGE (ADHCC210) -
    QUALIFIER (ADHTOOLS) -
    MEMBER (ADHAAAEN) -
    OWNER (PAOLOR1) -
    ACTION (REPLACE) -
    DYNAMICRULES (RUN) -
    EXPLAIN (NO) -
    ISOLATION (CS) -
    ENCODING (UNICODE) -
    VALIDATE (RUN)
*
    BIND PACKAGE (ADHCC210) -
        QUALIFIER (ADHTOOLS) -
        MEMBER (ADHAAFLR) -
        OWNER (PAOLOR1) -
        ACTION (REPLACE) -
        DYNAMICRULES (RUN) -
        EXPLAIN (NO) -
        ISOLATION (CS) -
        ENCODING (UNICODE) -
        VALIDATE (RUN)
*
        BIND PACKAGE (ADHCC210) -
            QUALIFIER (ADHTOOLS) -
            MEMBER (ADHTSQL) -
            OWNER (PAOLOR1) -
            ACTION (REPLACE) -
            DYNAMICRULES (RUN) -
            EXPLAIN (NO) -
            ISOLATION (CS) -
            ENCODING (UNICODE) -
            VALIDATE (RUN)
*
            BIND PACKAGE (ADHCC210) -
                QUALIFIER (ADHTOOLS) -
                MEMBER (ADHMSCM) -
                OWNER (PAOLOR1) -
                ACTION (REPLACE) -
                DYNAMICRULES (RUN) -
                EXPLAIN (NO) -
                ISOLATION (CS) -
                ENCODING (UNICODE) -
                VALIDATE (RUN)
*
                BIND PACKAGE (ADHCC210) -
                    QUALIFIER (ADHTOOLS) -
                    MEMBER (ADHMARM) -
                    OWNER (PAOLOR1) -
                    ACTION (REPLACE) -
                    DYNAMICRULES (RUN) -
                    EXPLAIN (NO) -
                    ISOLATION (CS) -
                    ENCODING (UNICODE) -

```

```

VALIDATE (RUN)
*
BIND PACKAGE (ADHCC210) -
  QUALIFIER (ADHTOOLS) -
  MEMBER (ADHMSUMT) -
  OWNER (PAOLOR1) -
  ACTION (REPLACE) -
  DYNAMICRULES (RUN) -
  EXPLAIN (NO) -
  ISOLATION (UR) -
  CURRENTDATA(NO) -
  ENCODING (UNICODE) -
  VALIDATE (RUN)
*
BIND PACKAGE (ADHCC210) -
  QUALIFIER (ADHTOOLS) -
  MEMBER (ADHMAAAC) -
  OWNER (PAOLOR1) -
  ACTION (REPLACE) -
  DYNAMICRULES (RUN) -
  EXPLAIN (NO) -
  ISOLATION (UR) -
  CURRENTDATA(NO) -
  ENCODING (UNICODE) -
  VALIDATE (RUN)
*
BIND PACKAGE (ADHCC210) -
  QUALIFIER (ADHTOOLS) -
  MEMBER (ADHMDLCP) -
  OWNER (PAOLOR1) -
  ACTION (REPLACE) -
  DYNAMICRULES (RUN) -
  EXPLAIN (NO) -
  ISOLATION (UR) -
  CURRENTDATA(NO) -
  ENCODING (UNICODE) -
  VALIDATE (RUN)
*
BIND PACKAGE (ADHCC210) -
  QUALIFIER (ADHTOOLS) -
  MEMBER (ADHASSCS) -
  OWNER (PAOLOR1) -
  ACTION (REPLACE) -
  DYNAMICRULES (RUN) -
  EXPLAIN (NO) -
  ISOLATION (UR) -
  CURRENTDATA(NO) -
  ENCODING (UNICODE) -
  VALIDATE (RUN)
*
BIND PACKAGE (ADHCC210) -
  QUALIFIER (ADHTOOLS) -
  MEMBER (ADHMADHM) -
  OWNER (PAOLOR1) -
  ACTION (REPLACE) -
  DYNAMICRULES (RUN) -
  EXPLAIN (NO) -
  ISOLATION (UR) -
  CURRENTDATA(NO) -
  ENCODING (UNICODE) -

```

```

        VALIDATE (RUN)
*
    BIND PACKAGE (ADHCC210) -
        QUALIFIER (ADHTOOLS) -
        MEMBER (ADHAAAIS) -
        OWNER (PAOLOR1) -
        ACTION (REPLACE) -
        DYNAMICRULES (RUN) -
        EXPLAIN (NO) -
        ISOLATION (UR) -
        CURRENTDATA(NO) -
        ENCODING (UNICODE) -
        VALIDATE (RUN)
END
/*
//

```

-
- ADHBND91—Bind Audit Management Expert packages for Log Analysis and data collection. See Example 9-9.

Example 9-9 ADHBND91: Bind Audit Management Expert packages for Log Analysis and data collection

```

//*****
/* Member: ADHBND91
/*
/* 5655-T57
/* Copyright Rocket Software, Inc. 2004 - 2008 All Rights
/* Reserved.
/*
//*****
/*
/* IBM DB2 Audit Management Expert for z/OS 2.1.0
/* This JCL binds product packages.
/* Use this JCL for DB2 version 9.
/*-----*
/*
/* Bind Step - Bind Product Packages
/*
//BIND1      EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB    DD DISP=SHR,DSN=DB9A9.SDSNLOAD
//SYSPRINT   DD SYSOUT=*
//SYSTSPRT   DD SYSOUT=*
//SYSUDUMP   DD SYSOUT=*
//DBRMLIB    DD DISP=SHR,DSN=ADH.V2R1MO.SADHDBRM
//SYSTSIN    DD *
            DSN SYSTEM(DB9A)
*
* ADH PRODUCT PACKAGES GO HERE
*
*
BIND PACKAGE (ADHCE210) -
    QUALIFIER (ADHTOOLS) -
    MEMBER (ADHZ020) -
    OWNER (PAOLOR1) -
    ACTION (REPLACE) -
    DYNAMICRULES (RUN) -
    EXPLAIN (NO) -
    ISOLATION (CS) -
    ENCODING (UNICODE) -
    VALIDATE (RUN)

```

```
    BIND PACKAGE (ADHCE210) -
      QUALIFIER (ADHTOOLS) -
      MEMBER (ADHZ021) -
      OWNER (PAOLOR1) -
      ACTION (REPLACE) -
      DYNAMICRULES (RUN) -
      EXPLAIN (NO) -
      ISOLATION (CS) -
      ENCODING (UNICODE) -
      VALIDATE (RUN)
```

*

* ASC SERVICES

*

```
    BIND PACKAGE (ADHCS210) -
      QUALIFIER (ADHTOOLS) -
      MEMBER (ADH@TBL8) -
      OWNER (PAOLOR1) -
      ACTION (REPLACE) -
      DYNAMICRULES (RUN) -
      EXPLAIN (NO) -
      ISOLATION (CS) -
      VALIDATE (RUN)
```

*

```
    BIND PACKAGE (ADHCS210) -
      QUALIFIER (ADHTOOLS) -
      MEMBER (ADH@SETP) -
      OWNER (PAOLOR1) -
      ACTION (REPLACE) -
      DYNAMICRULES (RUN) -
      EXPLAIN (NO) -
      ISOLATION (CS) -
      VALIDATE (RUN)
```

*

* LOG ANALYSIS SERVICES

*

```
    BIND PACKAGE (ADHCE310) -
      QUALIFIER (ADHTOOLS) -
      MEMBER (ADHSQL9) -
      OWNER (PAOLOR1) -
      ACTION (REPLACE) -
      DYNAMICRULES (RUN) -
      EXPLAIN (NO) -
      ISOLATION (CS) -
      ENCODING (EBCDIC) -
      VALIDATE (RUN)
```

*

```
    BIND PACKAGE (ADHCE310) -
      QUALIFIER (ADHTOOLS) -
      MEMBER (ADHSQL9A) -
      OWNER (PAOLOR1) -
      ACTION (REPLACE) -
      DYNAMICRULES (RUN) -
      EXPLAIN (NO) -
      ISOLATION (CS) -
      ENCODING (EBCDIC) -
      VALIDATE (RUN)
```

*

```
    BIND PACKAGE (ADHCE310) -
      QUALIFIER (ADHTOOLS) -
      MEMBER (ADHSQL9C) -
```



```

OWNER (PAOLOR1) -
ACTION (REPLACE) -
DYNAMICRULES (RUN) -
EXPLAIN (NO) -
ISOLATION (CS) -
ENCODING (EBCDIC) -
VALIDATE (RUN)
*
BIND PACKAGE (ADHCE310) -
  QUALIFIER (ADHTOOLS) -
  MEMBER (ADHSQ9CA) -
  OWNER (PAOLOR1) -
  ACTION (REPLACE) -
  DYNAMICRULES (RUN) -
  EXPLAIN (NO) -
  ISOLATION (CS) -
  ENCODING (EBCDIC) -
  VALIDATE (RUN)
BIND PACKAGE (ADHRP210) -
  QUALIFIER (ADHTOOLS) -
  MEMBER (ADH@USER) -
  OWNER (PAOLOR1) -
  ACTION (REPLACE) -
  DYNAMICRULES (RUN) -
  EXPLAIN (NO) -
  ISOLATION (CS) -
  ENCODING (EBCDIC) -
  VALIDATE (RUN)
BIND PACKAGE (ADHRP210) -
  QUALIFIER (ADHTOOLS) -
  MEMBER (ADH@OBJT) -
  OWNER (PAOLOR1) -
  ACTION (REPLACE) -
  DYNAMICRULES (RUN) -
  EXPLAIN (NO) -
  ISOLATION (CS) -
  ENCODING (EBCDIC) -
  VALIDATE (RUN)

END
/*
//

```

– For remote connection

- ADHBND93—Bind Audit Management Expert packages (for the repository).
- ADHBND94—Bind Audit Management Expert packages (for Log Analysis and data collection).

D- Granting access to packages

Grant the Audit Management Expert packages for each subsystem against which you plan to run DB2 Audit Management Expert. We installed Audit Management Expert on DB2 9, so we execute job ADHGRT9B only. See Example 9-10 on page 194.

Example 9-10 ADHGRT9B: Grant Audit Management Expert packages

```
//*****
/* IBM DB2 Audit Management Expert for z/OS 2.1.0
/*
/* 5655-T57
/* Copyright Rocket Software, Inc. 2004 - 2008 All Rights
/* Reserved.
/*
//*****
/* Use this JCL for DB2 version 9.
//ADHGRTA EXEC PGM=IKJEFT01,COND=(4,LT),DYNAMNBR=20
//STEPLIB DD DISP=SHR,DSN=DB9A9.SDSNLOAD
/*
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTEM DD SYSOUT=*
//CEEDUMP DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSTSIN DD *
    DSN SYSTEM(DB9A)
    RUN PROGRAM(DSNTEP2) PLAN(DSNTEP2) -
        LIB('DB9AU.RUNLIB.LOAD') PARMS('/ALIGN(MID)')
    END
//SYSIN DD *
--
-- IBM DB2 Audit Management Expert for z/OS table privilege grants
--
-- SET CURRENT SQLID = 'SYSADM';
--
GRANT EXECUTE ON PACKAGE ADHCC210.ADHMARM TO PAOLOR1;
GRANT EXECUTE ON PACKAGE ADHCC210.ADHAANAEN TO PAOLOR1;
GRANT EXECUTE ON PACKAGE ADHCC210.ADHAFLR TO PAOLOR1;
GRANT EXECUTE ON PACKAGE ADHCC210.ADHMARM TO PAOLOR1;
GRANT EXECUTE ON PACKAGE ADHCC210.ADHMSCM TO PAOLOR1;
GRANT EXECUTE ON PACKAGE ADHCC210.ADHMSUMT TO PAOLOR1;
GRANT EXECUTE ON PACKAGE ADHCC210.ADHTSQL TO PAOLOR1;
GRANT EXECUTE ON PACKAGE ADHCE210.ADHZ020 TO PAOLOR1;
GRANT EXECUTE ON PACKAGE ADHCE310.ADHSQ9CA TO PAOLOR1;
GRANT EXECUTE ON PACKAGE ADHCE310.ADHSQ9 TO PAOLOR1;
GRANT EXECUTE ON PACKAGE ADHCE310.ADHSQ9A TO PAOLOR1;
GRANT EXECUTE ON PACKAGE ADHCE310.ADHSQ9C TO PAOLOR1;
GRANT EXECUTE ON PACKAGE ADHRP210.ADH@OBJT TO PAOLOR1;
GRANT EXECUTE ON PACKAGE ADHRP210.ADH@USER TO PAOLOR1;
COMMIT;
-- If the inst#ADHASCSCer user id needs the privilege to bind plans,
-- uncomment the following lines and replace <installation user id> with
-- the user id requiring the privilege.
--GRANT BINDADD TO <installation user id> ;
--COMMIT;
/*
//*
```

E- Binding Audit Management Expert DB2 plans

Bind the Audit Management Expert plans for each subsystem against which you plan to run DB2 Audit Management Expert. We installed Audit Management Expert on DB2 9, so we execute job ADHBND92 only. See Example 9-11

Example 9-11 ADHBND92: Bind Audit Management Expert plans

```
//***** 00020000
//* 5655-T57 00040000
//* Copyright Rocket Software, Inc. 2004 - 2008 All Rights 00050000
//* Reserved. 00060000
//* 00070000
//***** 00080000
//* Plan Names Are: Plan #1 - ADHPLAN1 - Audit Management Expert 00342000
//* Plan #2 - ADHPLAN2 - Log Analysis 00343000
//* Plan #3 - ADHPLAN3 - Audit Management Expert Agent 00344000
//* Plan #4 - ADHPLAN4 - ASC Component Services 00344100
//* 00345000
//* The plan names used in this job must match the product plan names 00346000
//* defined in the product configuration for the target DB2 subsystem. 00347000
//* Sample JCL member ADHSJ001 is used to add/update product 00348000
//* configurations. 00349000
//*-----* 00360000
//* 00370000
//BIND EXEC PGM=IKJEFT01,DYNAMNBR=20 00380000
//STEPLIB DD DISP=SHR,DSN=DB9A9.SDSNLOAD 00390000
//SYSPRINT DD SYSOUT=* 00400000
//SYSPRINT DD SYSOUT=* 00410000
//SYSUDUMP DD SYSOUT=* 00420000
//DBRMLIB DD DISP=SHR,DSN=ADH.V2R1MO.SADHDBRM 00430000
//SYSTSIN DD * 00440000
DSN SYSTEM(DB9A) 00450000
00460000
* PLAN #1 ADH PRODUCT PLAN 00470000
  BIND PLAN (ADHPLAN1) - 00480000
    PKLIST (ADHCE210.*) - 00490000
      QUALIFIER (ADHTOOLS) - 00500000
      ACTION (REPLACE) - 00510000
      RETAIN - 00520000
      DYNAMICRULES (RUN) - 00530000
      EXPLAIN (NO) - 00540000
      ISOLATION (CS) - 00550000
      SQLRULES (DB2) - 00560000
      ENCODING (UNICODE) - 00561000
      VALIDATE (RUN) 00570000
00580000
* PLAN #2 LOG ANALYSIS SERVICES 00590000
  BIND PLAN (ADHPLAN2) - 00600000
    PKLIST (ADHCE310.*) - 00610000
      QUALIFIER (ADHTOOLS) - 00620000
      ACTION (REPLACE) - 00630000
      RETAIN - 00640000
      DYNAMICRULES (RUN) - 00650000
      EXPLAIN (NO) - 00660000
      ISOLATION (CS) - 00670000
      SQLRULES (DB2) - 00680000
      VALIDATE (RUN) 00690000
00700000
* PLAN #3 Agent 00710000
  BIND PLAN (ADHPLAN3) - 00720000
```

```

        PKLIST (*.ADHCC210.*, ADHRP210.*) - 00730000
        QUALIFIER (ADHTOOLS) - 00740000
        ACTION (REPLACE) - 00750000
        RETAIN - 00760000
        DYNAMICRULES (RUN) - 00770000
        EXPLAIN (NO) - 00780000
        ISOLATION (CS) - 00790000
        SQLRULES (DB2) - 00800000
        ENCODING (UNICODE) - 00801000
        VALIDATE (RUN) 00810000
                                00810100
* PLAN #4 ASC SERVICES 00811000
  BIND PLAN (ADHPLAN4) - 00812000
    PKLIST (*.ADHCS210.*) - 00813000
    QUALIFIER (ADHTOOLS) - 00814000
    ACTION (REPLACE) - 00815000
    RETAIN - 00816000
    DYNAMICRULES (RUN) - 00817000
    EXPLAIN (NO) - 00818000
    ISOLATION (CS) - 00819000
    SQLRULES (DB2) - 00819100
    VALIDATE (RUN) 00819200
                                00819300
                                00820000
END 00830000
/* 00840000
//* 00850000
// 00860000

```

E- Granting access to plans

Grant the Audit Management Expert plans for each subsystem which you plan to audit.

Note: It is not necessary for to run this step if the installation user ID was the same as the user ID that is used to run Audit Management Expert server, agent, and reporting. If it is run with all IDs being the same, it will fail, because a user ID cannot grant authority to itself.

Step 6: Configuring the server

- ▶ Modify member ADHCFGS the server configuration file to configure a connection to a local repository. See Example 9-12.

Example 9-12 ADHCFGS: Server configuration file

```

<server-config>

  <server-repository>DB9A</server-repository>
  <client-listener-port>52522</client-listener-port>
  <agent-listener-port>52521</agent-listener-port>
  <object-qualifier>ADHTOOLS</object-qualifier>
  <summarizer-refresh-interval>300</summarizer-refresh-interval>
  <trace-network>>false</trace-network>
  <trace-events>>false</trace-events>
  <trace-config>>true</trace-config>

</server-config>

```

- ▶ Modify and run member ADHSJSRV the Audit Management Expert server JCL. See Example 9-13 on page 197.

Example 9-13 ADHSJSRV—Audit Management Expert server JCL

```
//ADHSERV PROC RGN=OM,  
// TME=1440,  
// PRM=,  
// LOAD=ADH.V2R1MO.SADHLOAD,  
// PCF=ADH.V2R1MO.CONTROL,  
// CFG=ADH.V2R1MO.SADHSAMP.CUSTOM(DB9ACFGS)  
/*  
//ADHSRV EXEC PGM=ADHSAES,REGION=&RGN,TIME=&TME,PARM=&PRM  
//STEPLIB DD DISP=SHR,DSN=&LOAD  
//ADHCFG DD DISP=SHR,DSN=&CFG  
//DB2PARMS DD DISP=SHR,DSN=&PCF  
//ADHLOG DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//SYSOUT DD SYSOUT=*  
//CEEDUMP DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*  
// PEND
```

- ▶ After starting the Audit Management Expert server, verify that the server and repository have been properly configured by inspecting the ADHLOG DD of the server for error messages.

Step 7: Configuring the ASC

- ▶ Edit and run member ADHINTER where ASC is installed to creating the ADHINTER data set. See Example 9-14.

Note: Interval data sets must not be shared in a data sharing environment. Each ASC subsystem requires the creation of its own unique interval data set.

Example 9-14 ADHINTER: Creating ADHINTER dataset

```
/*  
/* NAME = ADHINTER *  
/* *  
/* 5655-T57 *  
/* Copyright Rocket Software, Inc. 1999 - 2008 All Rights Reserved. *  
/* *  
/* DESCRIPTION: THIS JCL USED TO ALLOCATE *  
/* THE INTERVALS DATASET FOR *  
/* IBM AUDIT MANAGEMENT EXPERT AUDIT SQL COLLECTOR V2.1 *  
/* *  
/STEP1 EXEC PGM=IDCAMS  
/SYSPRINT DD SYSOUT=*  
/SYSIN DD *  
DELETE ADH.SC63.INTERVAL CLUSTER  
IF LASTCC<=8 THEN SET MAXCC=0  
DEFINE CLUSTER(NAME(ADH.SC63.INTERVAL) -  
VOL(SBOXB4) -  
CYLINDERS(1 1) -  
RECORDSIZE(12 32760) -  
CISZ(32768) -  
SHAREOPTIONS(2,3) -  
INDEXED UNIQUE -  
KEYS(12 0) -  
) -  
DATA(NAME(ADH.SC63.INTERVAL.DATA)) -  
INDEX(NAME(ADH.SC63.INTERVAL.INDEX))
```

► Tailoring the ADHCFGP data set

The ADH#MAIN program uses parameters. These parameters are defined in an 80-byte sequential or partitioned data set that the user must allocate to the ADHCFGP DD. A sample is located in SADHSAMP library member ADHCFGP. See Example 9-15.

Example 9-15 ADHCFGP: ADH#MAIN program parameters

```

SUBSYS(DB9A)                -
  AUDIT_HOSTV_DSN(ADHUSER.DB9A.AHSTV.D&LYMMDD.&INTV.) -
  AUDIT_TEXT_DSN(ADHUSER.DB9A.ATEXT.D&LYMMDD.&INTV.) -
  AUDIT_STATEMENT_DSN(ADHUSER.DB9A.ASTMT.D&LYMMDD.&INTV.) -
  AUDIT_OBJECTS_DSN(ADHUSER.DB9A.AOBS.D&LYMMDD.&INTV.) -
  AUDIT_HOSTV_SPACE_UNITS(CYLS) -
  AUDIT_TEXT_SPACE_UNITS(CYLS) -
  AUDIT_STATEMENT_SPACE_UNITS(CYLS) -
  AUDIT_OBJECTS_SPACE_UNITS(CYLS) -
  AUDIT_HOSTV_PRIMARY(05) -
  AUDIT_TEXT_PRIMARY(05) -
  AUDIT_STATEMENT_PRIMARY(05) -
  AUDIT_OBJECTS_PRIMARY(05) -
  AUDIT_HOSTV_SECONDARY(05) -
  AUDIT_TEXT_SECONDARY(05) -
  AUDIT_STATEMENT_SECONDARY(05) -
  AUDIT_OBJECTS_SECONDARY(05) -
  DEBUG(N) -
  AUTHID(PAOLOR1) -
  INTERVAL_MIDNIGHT(N) -
  OBJECTS(Y) -

```

► Defining the ASC started task JCL

The ASC runs as a started task. The member ADHCSSID contains the sample JCL to setup the Audit Management Expert ASC started task. The started task should be named ADHCSSID, where *SSID* is the identifier of the DB2 Subsystem to be monitored. See Example 9-16.

Example 9-16 ADHCDB9A: ASC started task

```

//ADHCSSID PROC RGN=7M,                                00006000
// LOAD=ADH.V2R1M0.SADHLOAD,                          00006100
// FECLoad=FEC.V2R1M0.SFECLoad,                      00006200
// PCF=ADH.V2R1M0.CONTROL,                            00006300
// CFG=ADH.V2R1M0.SADHSAMP.CUSTOM(DB9ACFGP),          00006401
// INTER=ADH.SC63.INTERVAL                            00006500
//*                                                    00007000
//ADHCSSID EXEC PGM=ADH#MAIN,REGION=&RGN,DYNAMNBR=200,TIME=1440 00013400
//STEPLIB DD DSN=&LOAD,DISP=SHR                       00013600
//          DD DSN=&FECLoad,DISP=SHR                   00013700
//DB2PARMS DD DSN=&PCF,DISP=SHR                       00014000
//ADHPARMS DD DSN=&CFG,DISP=SHR                       00014100
//ADHINTER DD DSN=&INTER,DISP=SHR                    00015000
//SYSPRINT DD SYSOUT=*                               00015100
//SYSUDUMP DD SYSOUT=*                               00016000

```

Step 8: Configuring the agent

► Configuring a connection to a local repository

- Create an agent configuration file using ADHCFGA sample agent configuration as an example. See Example 9-17.

Example 9-17 DB9ACFGA

```
<agent-config>
  <server-address>wtsc63.itso.ibm.com</server-address>
  <agent-monitor>DB9A</agent-monitor>
  <server-repository>DB9A</server-repository>
  <object-qualifier>ADHTOOLS</object-qualifier>
  <object-collection>ADHCC210</object-collection>
  <server-port>52521</server-port>
  <log-level>I</log-level>
  <trace-ifi>>false</trace-ifi>
  <trace-network>>false</trace-network>
  <trace-db2-attachment>>false</trace-db2-attachment>
  <trace-sql>>false</trace-sql>
  <trace-db2loadstore>>false</trace-db2loadstore>
  <trace-db2load>>false</trace-db2load>
  <trace-norm>>false</trace-norm>
  <trace-xml>>false</trace-xml>
  <trace-events>>false</trace-events>
  <trace-csi>>false</trace-csi>
  <trace-config>>true</trace-config>
</agent-config>
```

- Create and run a Audit Management Expert agent JCL using ADHSJAGT sample. See Example 9-18.

Example 9-18 ADHDB9AA: Audit Management Expert agent JCL

```
//ADHDB9AA PROC RGN=0M,
// TME=1440,
// PRM=,
// LOAD=ADH.V2R1MO.SADHLOAD,
// FECLOAD=FEC.V2R1MO.SFECLOAD,
// MENU=ADH.V2R1MO.SADHMENU,
// SLIB=ADH.V2R1MO.SADHSLIB,
// PCF=ADH.V2R1MO.CONTROL,
// CFG=ADH.V2R1MO.SADHSAMP.CUSTOM(DB9ACFGA)
//*
//ADHDB9AA EXEC PGM=ADHAAEA,REGION=&RGN,TIME=&TME,PARAM=&PRM
//STEPLIB DD DISP=SHR,DSN=&LOAD
// DD DISP=SHR,DSN=&FECLOAD
//ADHLOAD DD DISP=SHR,DSN=&LOAD
//ADHMLIB DD DISP=SHR,DSN=&MENU
//ADHSLIB DD DISP=SHR,DSN=&SLIB
//DB2PARMS DD DISP=SHR,DSN=&PCF
//ADHCFG DD DISP=SHR,DSN=&CFG
//ADHLOG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//CEEDUMP DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
```

Step 9: User administration procedure

Password validation is a key feature of Audit Management Expert security. The DB2 Audit Management Expert for z/OS user administration procedure (UAP) can be used to create a new user that has administrator privileges.

Do not use this procedure as a substitute for general user administration through the administration interface (for example, to update passwords for existing users). This procedure should only be used in those situations where you need to create a new user ID or to reset a password.

Use the DB2 Audit Management Expert for z/OS UAP to create a new user that has administrator privileges (a secure method of assigning an initial administrator ID and password).

The user administration procedure consists of the following components:

- ▶ SAMPLIB(ADHCFGU)—The configuration file for input to the ADHSUAP executable.
- ▶ LOADLIB(ADHSUAP)—The executable.
- ▶ SAMPLIB(ADHSJUAP) —The JCL you need to submit to invoke the module ADHSUAP.

The Audit Management Expert UAP executable runs as a batch job under z/OS. It is a manually submitted job.

UAP Configuration file

The Audit Management Expert UAP requires a configuration file for input, A sample configuration file (with the minimum options specified) is provided in the sample library member ADHCFGU. Copy this member and customize it. See Example 9-19.

Example 9-19 ADHCFGU: Audit Management Expert UAP configuration file

```
<uap-config>
<adh-user>adhadmin</adh-user>
  <new-adh-user-password>redbook1</new-adh-user-password>
  <server-repository>DB9A</server-repository>
  <object-qualifier>ADHTOOLS</object-qualifier>
  <object-collection>ADHCC210</object-collection>
</uap-config>
```

Audit Management Expert UAP JCL

The member ADHSJUAP contains sample JCL to run the Audit Management Expert UAP. See Example 9-20.

Example 9-20 ADHSJUAP Audit Management Expert UAP JCL

```
//ADHSUAP EXEC PGM=ADHSUAP,REGION=0M,TIME=1440
//STEPLIB DD DISP=SHR,DSN=ADH.V2R1MO.SADHLOAD
//ADHCFG DD DISP=SHR,DSN=ADH.V2R1MO.SADHSAMP.CUSTOM(ADHCFGU)
//DB2PARMS DD DISP=SHR,DSN=ADH.V2R1MO.CONTROL
//ADHLOG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//CEEDUMP DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
```

Creating a user

This information describes how to create a new user that has administrative privileges.

- ▶ Modify the configuration file, ADHCFGU, to specify a user ID for the <adh-user> and specify a password for the <new-adh-user-password>
- ▶ Modify and run the UAP JCL (ADHSJUAP) to create the new user ID with specified password

Resetting a password

This information describes how to reset a password.

- ▶ Modify the configuration file, ADHCFGU, to specify a user ID of an existing user and specify a password for the <new-adh-user-password>
- ▶ Modify and run the UAP JCL (ADHSJUAP) to reset password to the password specified by <new-adh-user-password>.

Deleting the password from the UAP configuration file

After running the UAP job and verifying the password, for security purposes, delete the new password from the configuration file ADHCFGU.

- ▶ Verify that the password by successfully logging into Audit Management Expert administration interface
- ▶ Remove the password from the member ADHCFGU.

Step 10: Binding the JDBC driver

You must run the bind for db2.jcc.DB2Driver on the DB2 subsystem that contains the repository prior to running the reporting client.

A sample command is as follows:

```
java com.ibm.db2.jcc.DB2Binder -url jdbc:db2:/// -user -password -action replace
```

Note: As of DB2, UDB Universal JDBC driver requires a license JAR file to be in the CLASSPATH along with the db2jcc.jar file db2jcc_license_cisuz.jar.

Step 11: Enabling audit data collection for reporting

Configure JDBC connection information to enable the reporting client to access audit data in the repository.

- ▶ Log in to the administration client.
- ▶ Edit the JDBC connection information.
- ▶ Using the administration client, set the following required parameters:
 - DB2LOAD Dataset HLQ to where input data sets for DB2LOAD utility are created.
 - DB2LOAD JCL Job Card.
- ▶ From the administration interface, Authorizations tab, authorize the reporting users to enable these users to view reports.

Step 12: Validating reporting access

After you configure the JDBC information, validate the connection to the repository from the reporting interface.

- ▶ Log in to the reporting client.
- ▶ Validate that the user ID has access to the subsystem.

Step 13: Installing the administration and reporting GUI clients

The GUI client installation is a member of a data set from the z/OS installation.

1. Use FTP to transfer member adhh1q.SADHGUIW(ADHGUIW), in binary to your workstation
2. Rename the member with a local file name of ADHGUIW.zip.
3. Extract ADHGUIW.zip. We will have two files: AuditManagementExpertReportingV2.exe and AuditManagementExpertAdminV2.exe.

Installing the administration GUI

- Open file AuditManagementExpertAdminV2.exe and the Welcome to the installer Audit Management Expert administration will appear. See Figure 9-4.

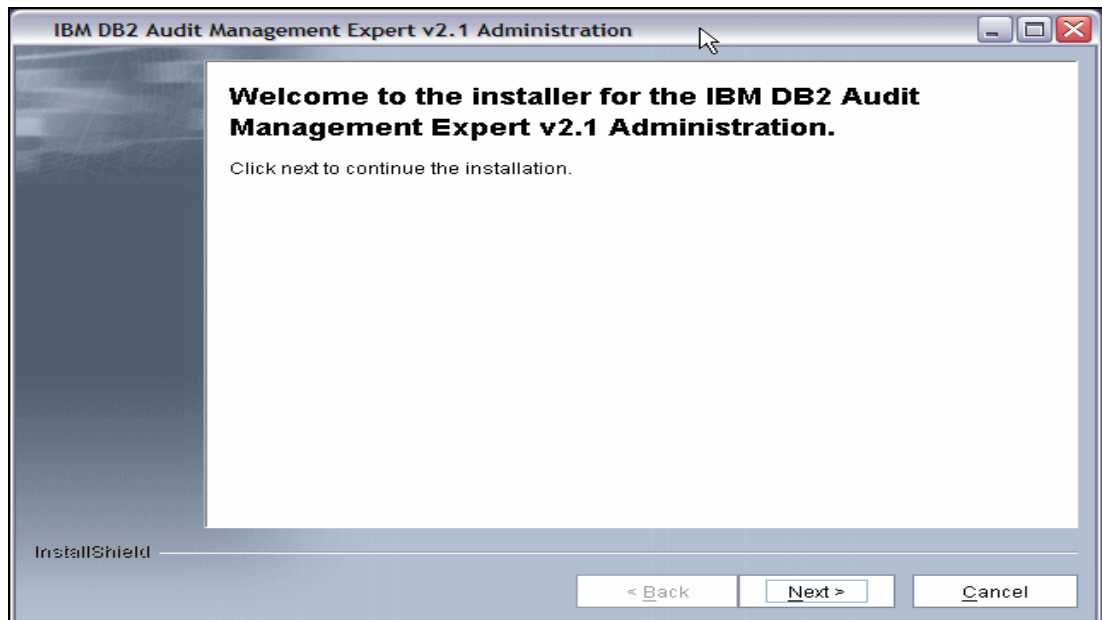


Figure 9-4 Audit Management Expert Administration: Welcome

- ▶ Choose a directory to install and click **Next**. See Figure 9-5.

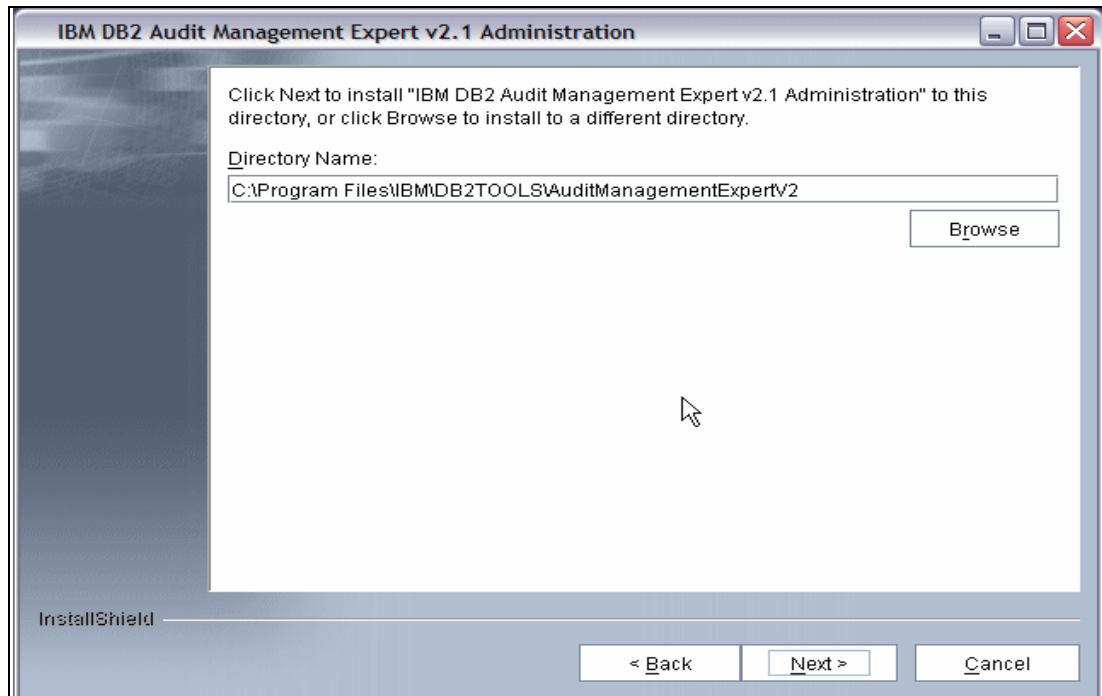


Figure 9-5 Audit Management Expert Administration: Choose Directory

- ▶ Select the **Yes** radio button to create a desktop shortcut. See Figure 9-6.

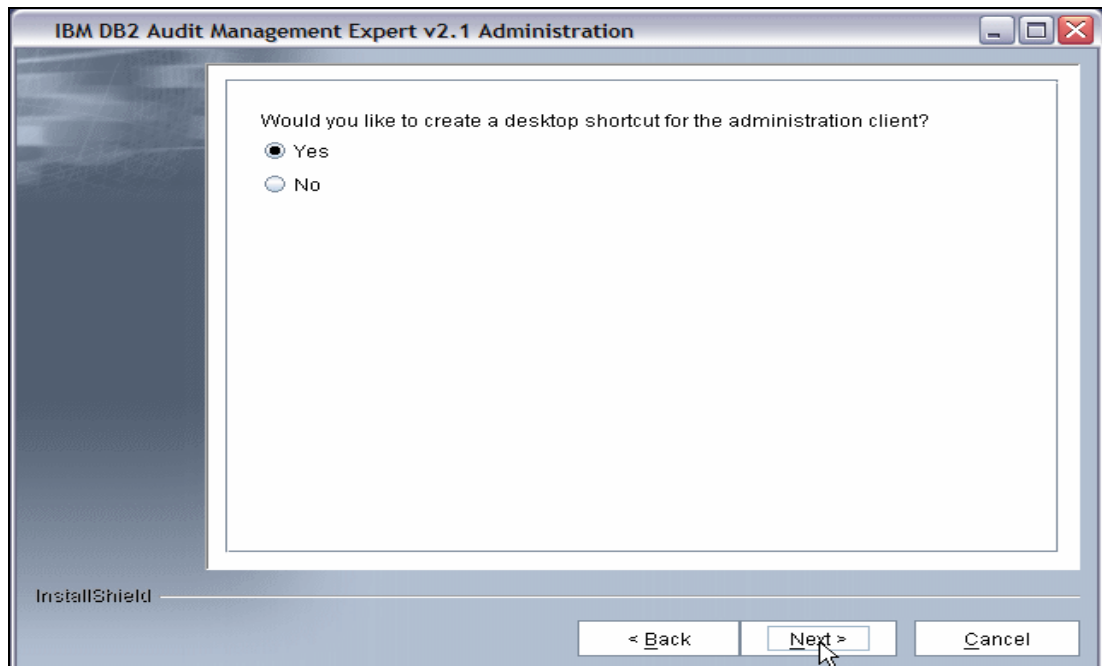


Figure 9-6 Audit Management Expert Administration: Shortcut

- Figure 9-7 is an installation summary. Click **Next**.

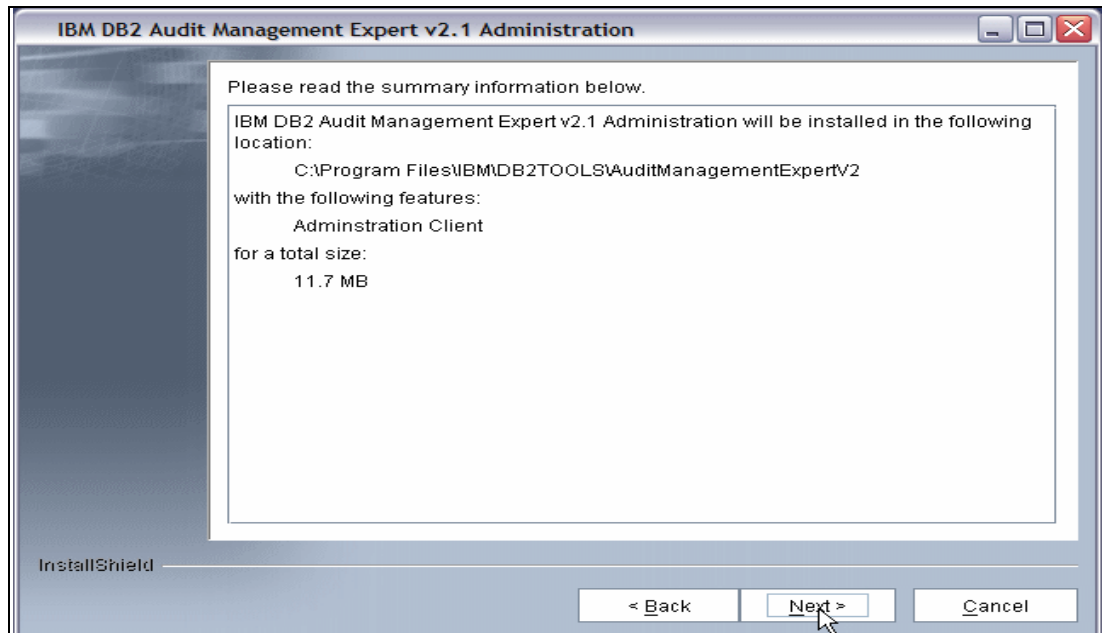


Figure 9-7 Audit Management Expert Administration: Installation Summary

- Installation in progress. See Figure 9-8.

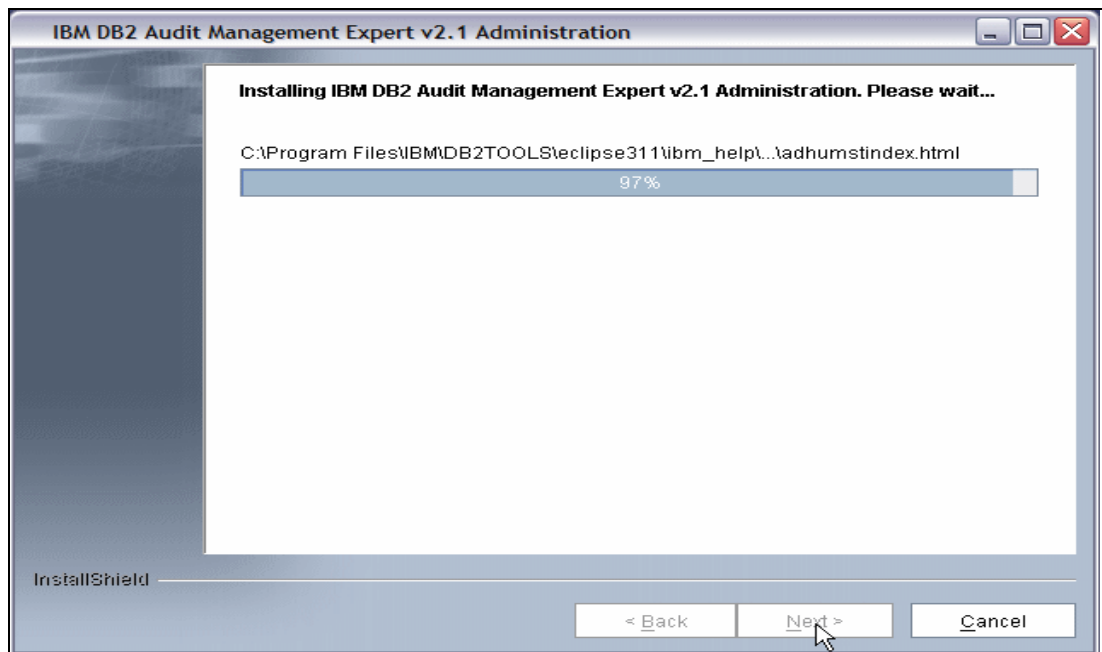


Figure 9-8 Audit Management Expert Administration: Installing

- ▶ Installation successful. Click **Finish**. See Figure 9-9.

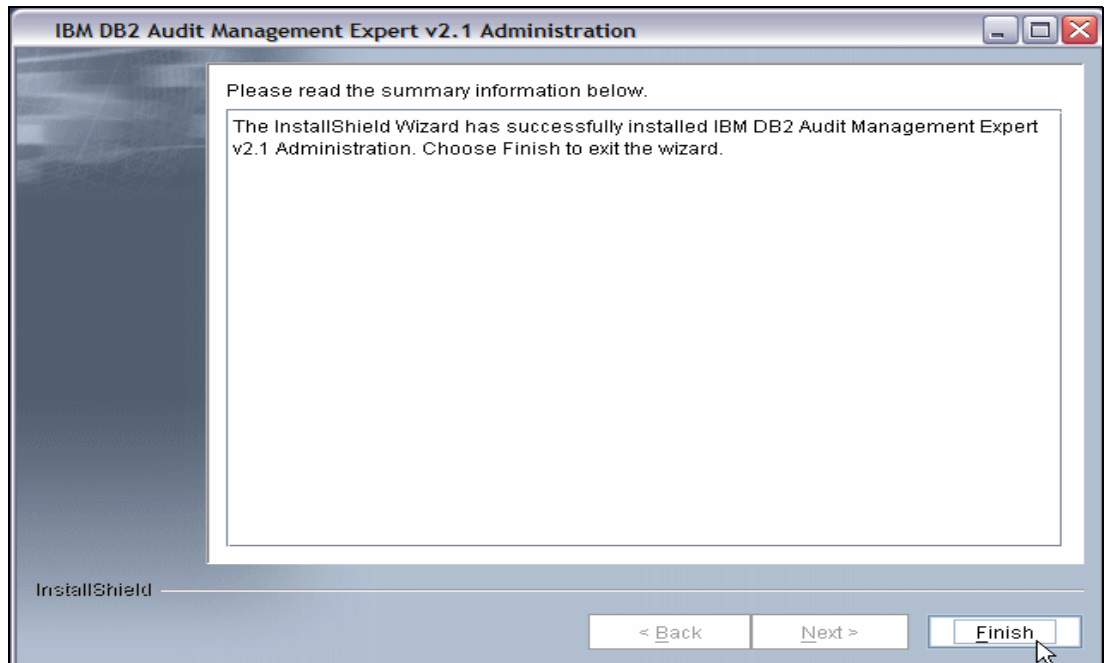


Figure 9-9 Audit Management Expert Administration: Installation Finish

- ▶ Click the **IBM DB2 Audit Management Expert V2.1 Admin** icon on your desktop.
- ▶ Click **Setting** → **Define Servers**. See Figure 9-10.

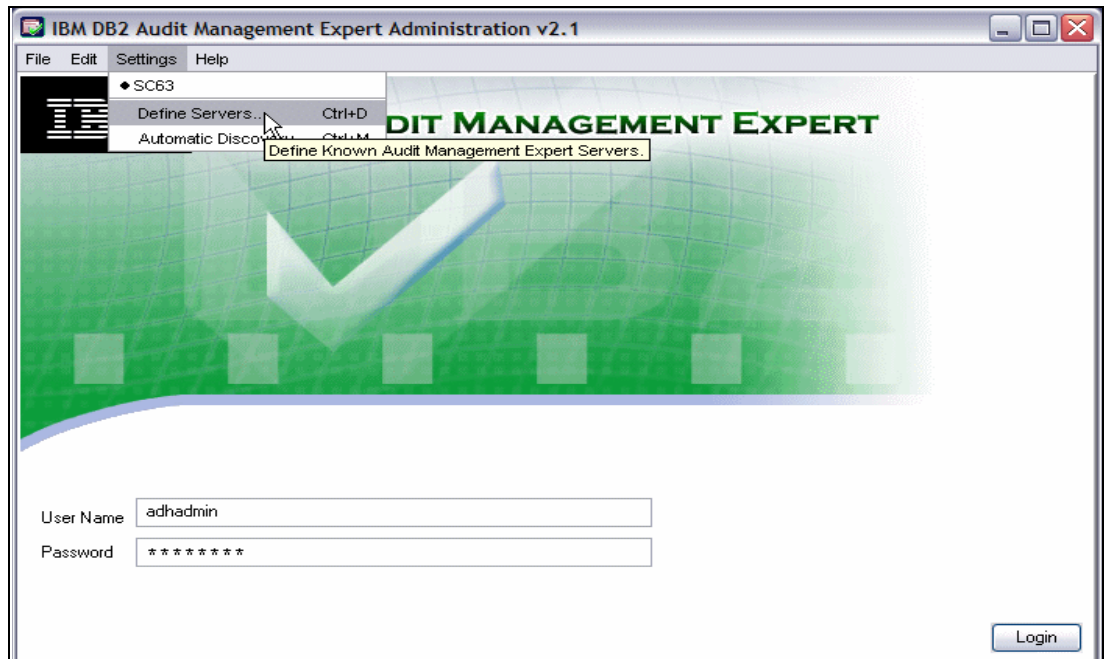


Figure 9-10 Audit Management Expert Administration: Defining Server

- ▶ Figure 9-11 displays our server definition.

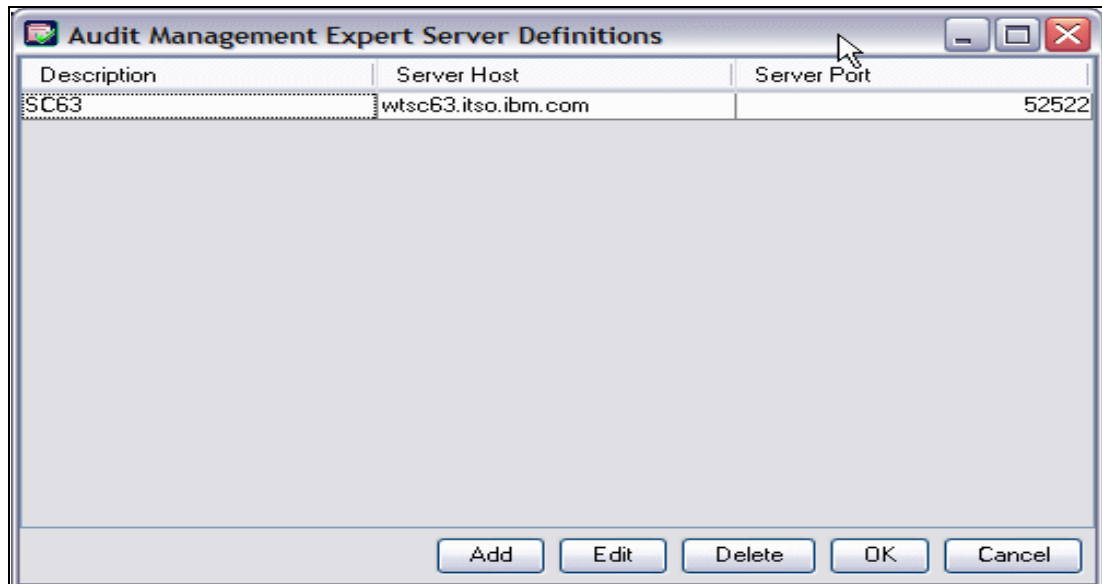


Figure 9-11 Audit Management Expert Administration: Defining Server

- ▶ Log on to the Audit Management Expert Administration. See Figure 9-12.



Figure 9-12 Audit Management Expert Administration: Logging in

Installing the reporting GUI

- ▶ Open file AuditManagementExpertReportingV2.exe and the “Welcome to the installer Audit Management Expert Reporting” panel displays. See Figure 9-13.

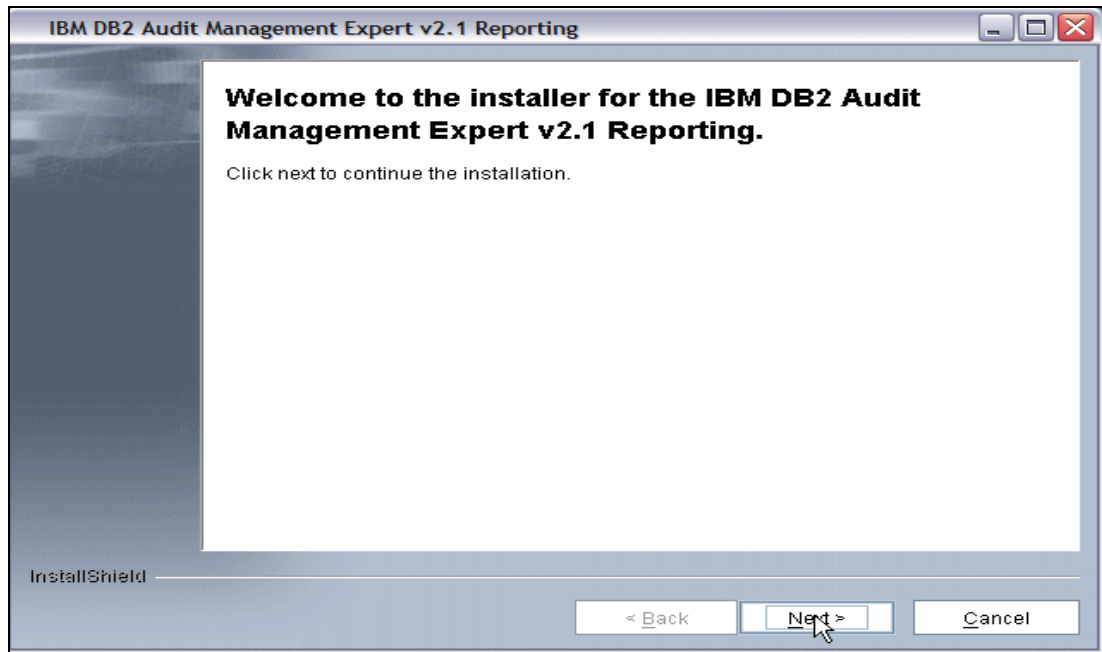


Figure 9-13 Audit Management Expert Report: Welcome

- ▶ Choose a directory to install and click **Next**. See Figure 9-14.

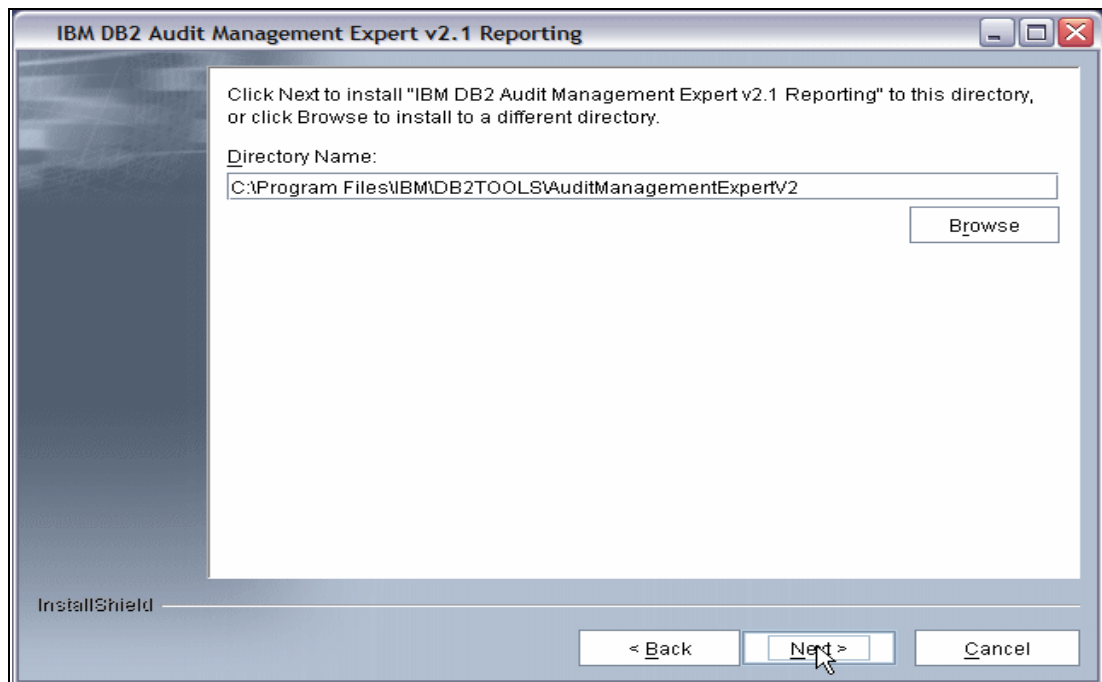


Figure 9-14 Audit Management Expert Report: Choose Directory

- ▶ Select the **Yes** radio button create a desktop shortcut. See Figure 9-15.

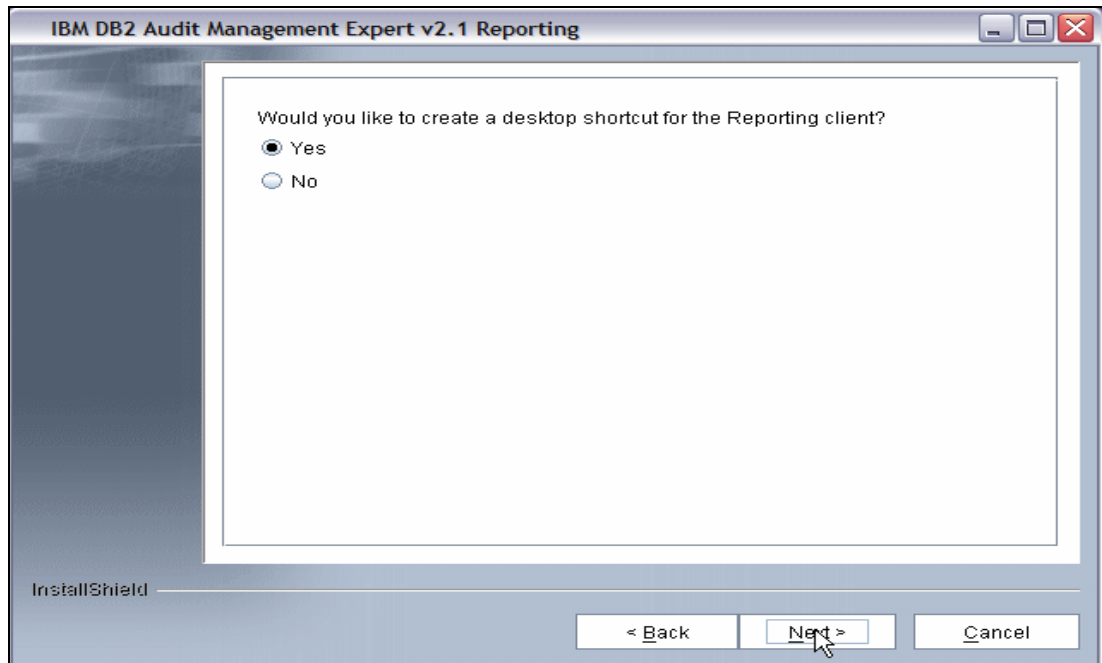


Figure 9-15 Audit Management Expert Report: Create Shortcut

- ▶ Figure 9-16 displays a summary of installation. Click **Next**.

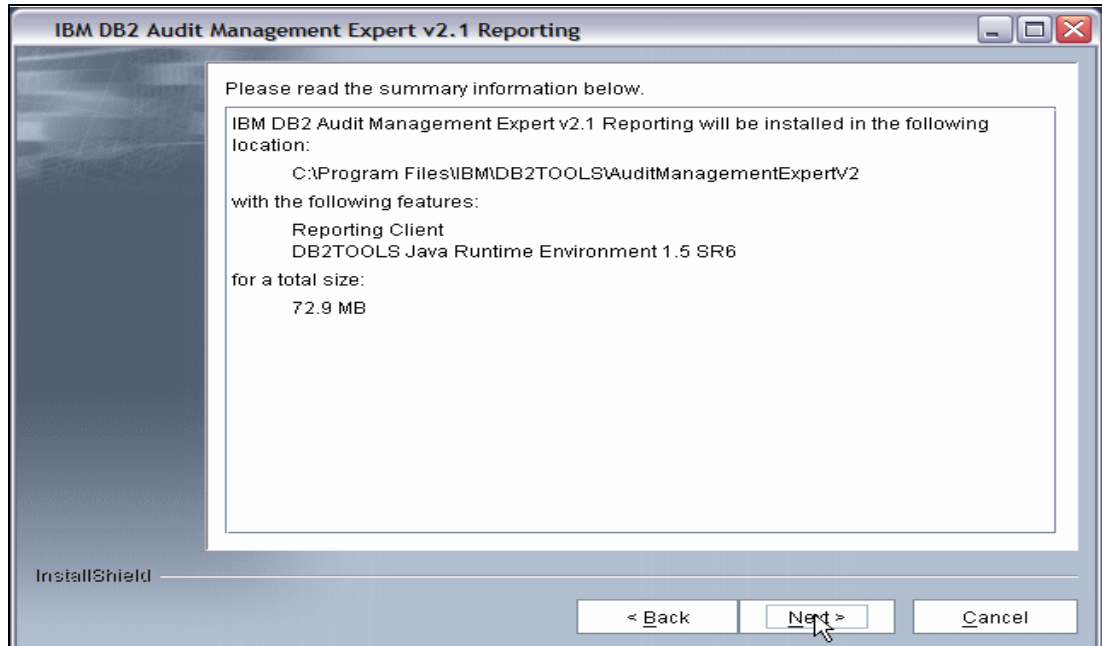


Figure 9-16 Audit Management Expert Report: Installation summary

- Installation successful. Click **Finish**. See Figure 9-17.

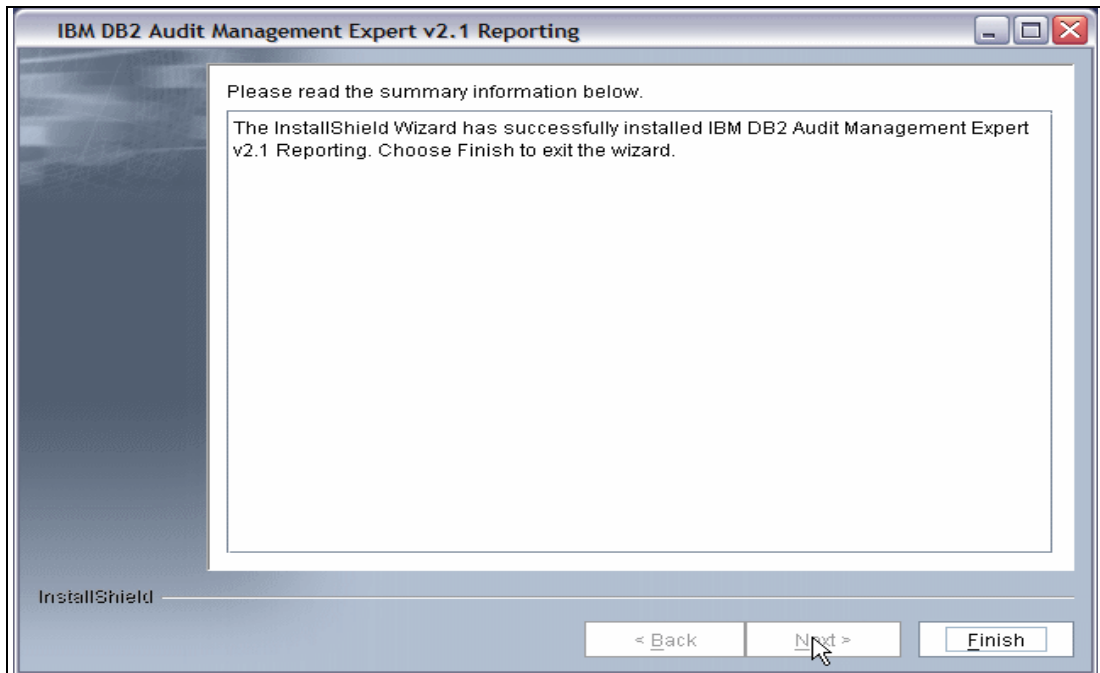


Figure 9-17 Audit Management Expert Report: Installation finish

- Click the **IBM DB2 Audit Management Expert V2.1 Reporting** icon on your desktop.
- Click **Setting** → **Define Servers**. See Figure 9-18.

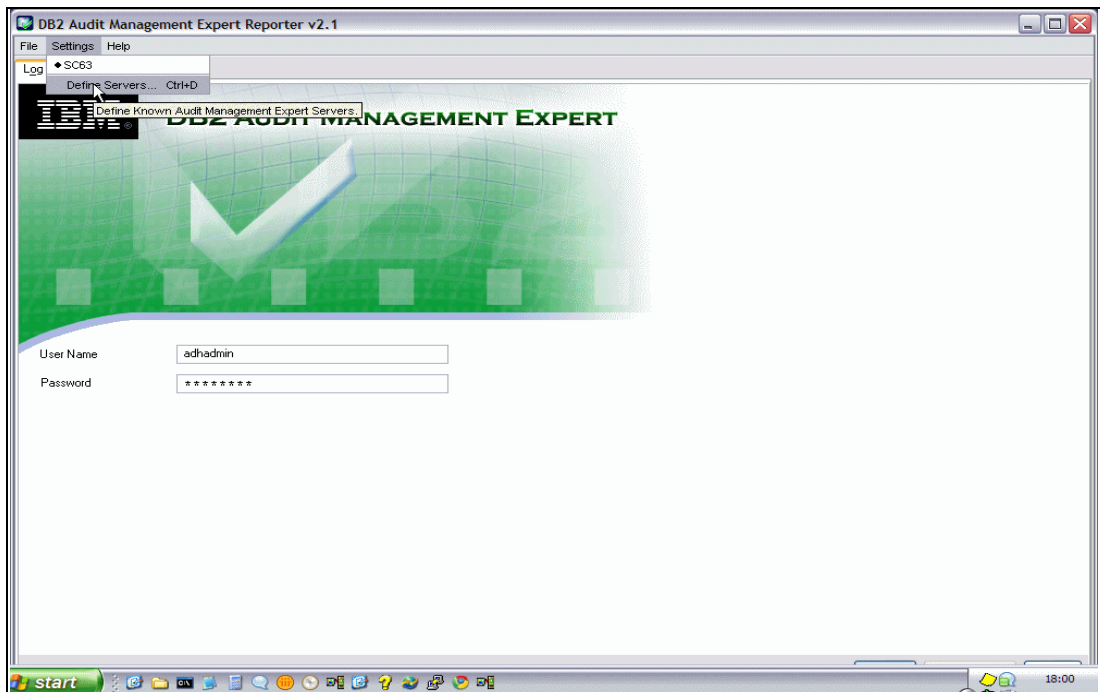


Figure 9-18 Audit Management Expert Report: Defining Server entry

- ▶ Figure 9-19 displays our server definition.



Figure 9-19 Audit Management Expert Report: Defining Server value

- ▶ Log on to Audit Management Expert reporting. See Figure 9-20.

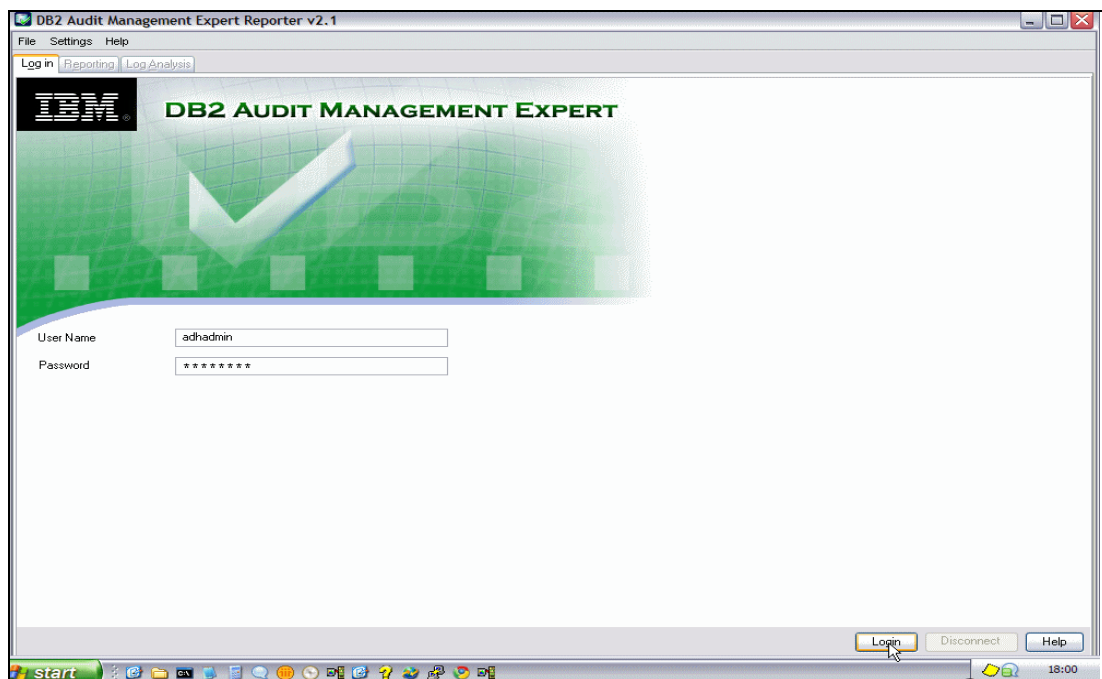


Figure 9-20 Audit Management Expert Report: Log in



Audit Management Expert scenarios

In this chapter we show the most common functions and provide some typical usage scenarios when utilizing the Audit Management Expert.

We first look at the administration functions of Audit Management Expert using the Administration User Interface to define different classes of auditing responsibilities, and to create collection profiles and collection definitions. Then we look at Reporter User Interface and provide some user scenarios to show the applicability of the reporter and its powerful functions for auditing needs.

The Administration User Interface is usually managed by the lead auditor. It sets up administration items such as user ID's and authorizations which provides the ability to assign auditor's access to the tool which in turn allows them access to the repository data. It is also used to set agent settings, repository information, create collection profiles and active or disable collections.

The Reporter User Interface enables the auditors to view and report on the audit data collected and has two components:

- ▶ The Reporting User Interface, which enables auditors to view audit data and generate audit reports.
- ▶ The Log Analysis User Interface, which enables auditors to generate reports using the log analysis function

For these reasons, DB2 Audit Management Expert for z/OS makes auditing data much more simple and manageable.

In the next several sections, we present scenarios which illustrate the functions to satisfy the following auditing requirements:

- ▶ Defining audit responsibilities
- ▶ Auditing privileged users
- ▶ Finding all authorization failures
- ▶ Finding DDL activity
- ▶ Tracking "who modified what"

10.1 Defining audit responsibilities

The administration client component of DB2 Audit Management Expert enables security administrators to perform a variety of administrative tasks, including managing users, groups, authorizations, collection profiles, collections, agents, and the repository.

In this section, we describe how to use the administration client to manage users and groups and enable different classes of auditing responsibilities.

Start the Audit Management Expert administration client from the Windows Start Menu. You will see the login panel as shown in Figure 10-1. To log in to Audit Management Expert from the administration client, specify the Audit Management Expert server to which you want to connect, and enter the user name and password.

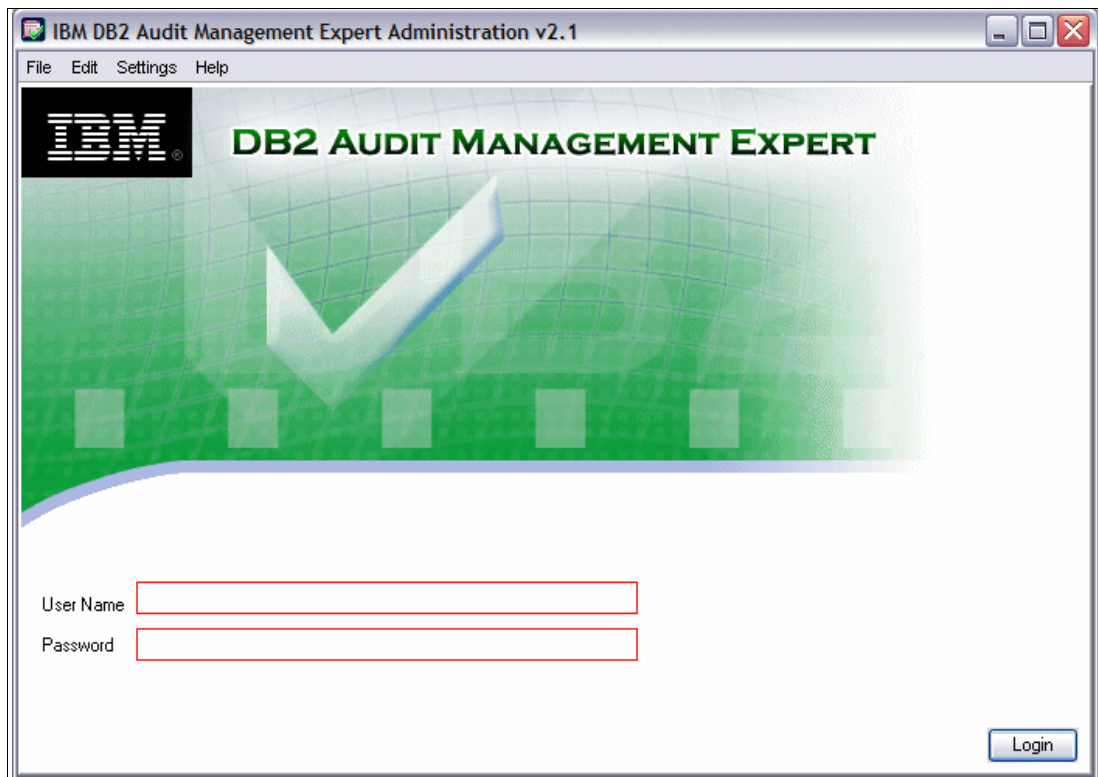


Figure 10-1 DB2 Audit Management Expert Login

The first time you log in, specify the user name and password of the default administrator.

Note: The default administrator is created using the user administration procedure (UAP), which runs as a batch job under z/OS. It is a secure method of assigning an initial administrator ID and password. The manually submitted job is included in DB2 Audit Management Expert's SAMPLIB, with name ADHSJUAP.

From the administration client Users tab (shown in Figure 10-2), administrators can create and assign permissions to users, edit users, copy users, and delete users. For each user, the Users tab displays information to identify the user and the user's assigned privileges. A checkmark indicates that the user has been assigned the corresponding privilege.

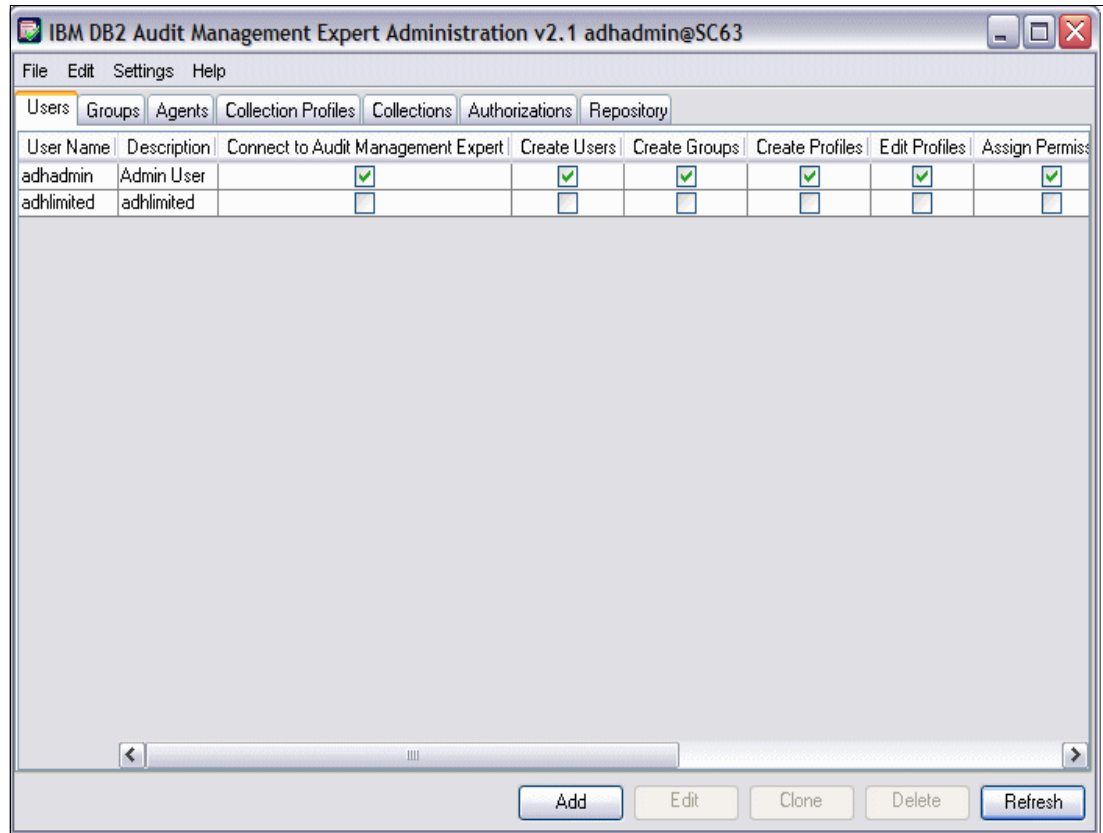


Figure 10-2 User administration list

From the Users tab, click **Add** to open the New User Wizard (Figure 10-3). On the left of the page you find the outline of the steps needed to add a new user to DB2 Audit Management Expert.

First, specify the user name and password of the new user. In our scenario, we created a user with the name PAOLOR3, who might be a new auditor assigned to a case. You can set an optional description or comment for the new user. Also, you can set an expiration date for the user account. This feature allows you to avoid having someone holding privileges when no longer required.

The screenshot shows a 'New User Wizard' dialog box. On the left is a sidebar with a tree view containing 'User Name', 'Permissions', 'Groups', and 'Summary', with 'User Name' selected. The main area contains the following fields and text:

- User Name: PAOLOR3
- Password: *****
- Confirm password: *****
- Description: User for Redbook Project
- Expires in: never [dropdown arrow] days.

Below the fields is a text block: "This grants the user the ability to utilize the Audit Management Expert system along with additional privileges granted to them or to the groups to which they are assigned. An optional description or comment can also be provided. The password will expire after the indicated number of days, and the user must provide a new one."

At the bottom right are three buttons: 'Next >', 'Finish', and 'Cancel'.

Figure 10-3 New user wizard

Click **Next** to begin assigning user permissions (Figure 10-4). Permissions enable or prevent users from performing specific activities within DB2 Audit Management Expert.

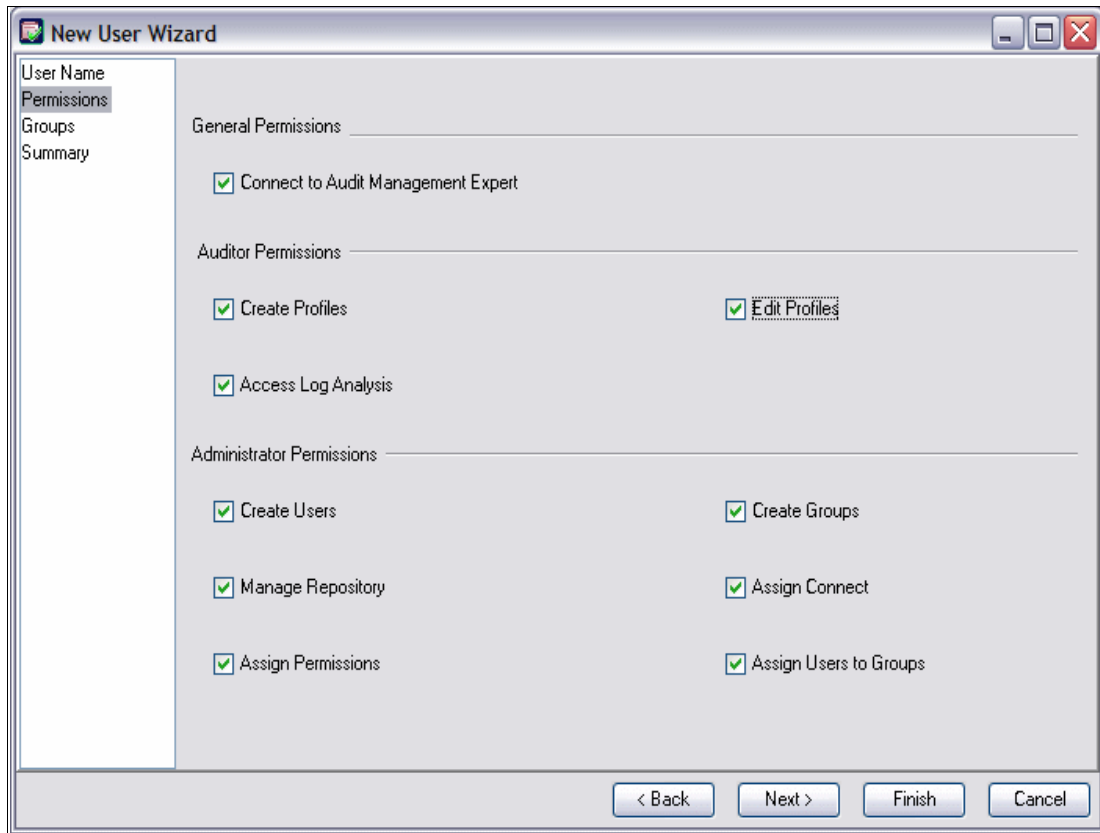


Figure 10-4 User permissions

Figure 10-4 on page 215 shows three types of permissions, each of which provides one or several privileges:

- ▶ **General Permissions**

Users assigned General Permissions can only connect to Audit Management Expert and view the auditing report.

- ▶ **Auditor Permissions**

Users assigned Auditor Permissions have more privileges, such as creating auditing profiles, editing auditing profiles, and accessing log analysis function.

- ▶ **Administrator Permissions**

Administrator Permissions are usually designated to administrators because they allow for functions that allow user, group, and repository management.

To assign a privilege to the user, select the check box that corresponds to the privilege you want to assign to that user. To remove a privilege, clear the check box. In our scenarios, we granted all privileges to user PAOLOR3.

Click **Next** to assign the user to one or multiple user groups (Figure 10-5).

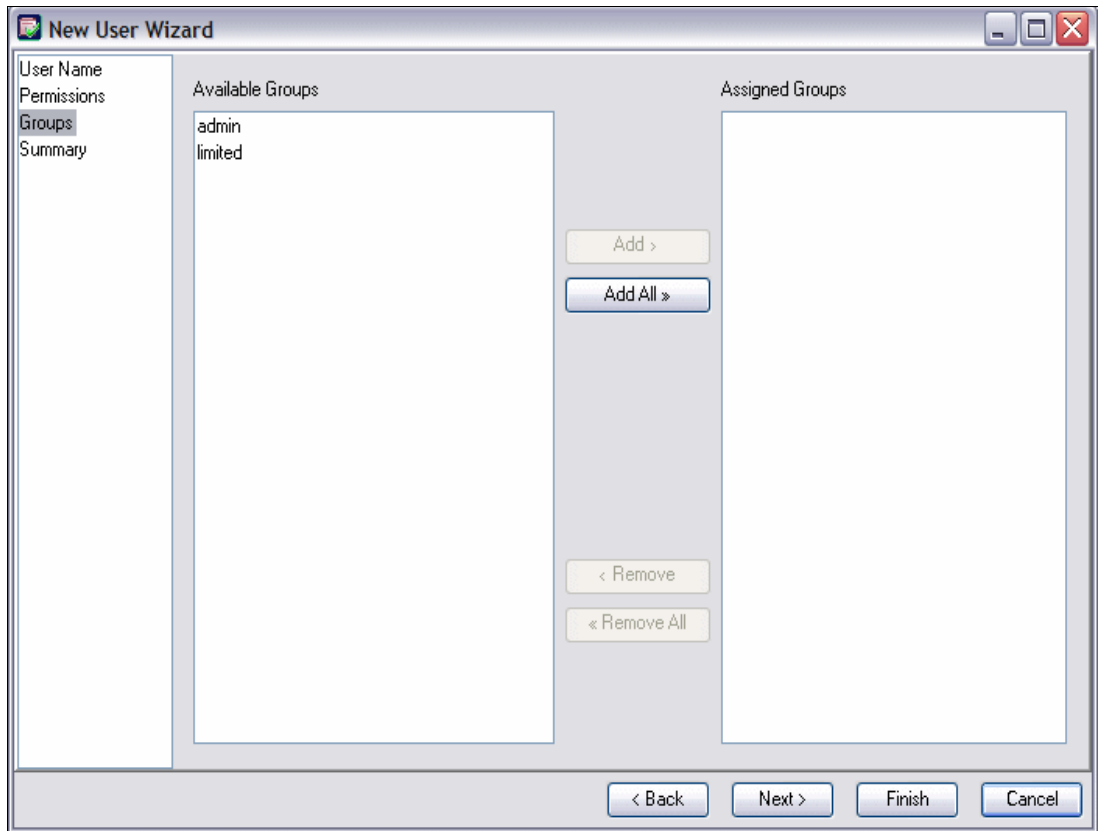


Figure 10-5 User group assignments

Assigning users to groups is an easy method of assigning a common set of privileges to more than one user. Available groups to which the user can be assigned are shown in the Available Groups list. Groups that the user is currently a member of are shown in the Assigned Groups list. To assign the user to a group, select the group in the Available Groups list to which you want to add the user and click **Add**. Similarly, you can remove the user from a group. To manage groups, click the **Groups** tab in Figure 10-2 on page 213.

Click **Next** again to see the User Summary of PAOLOR3 (Figure 10-6). A User Summary of the newly created user is the last window in the New User Wizard. The summary includes the new user name, description, encrypted password, expiration date, group assignments, and assigned privileges.

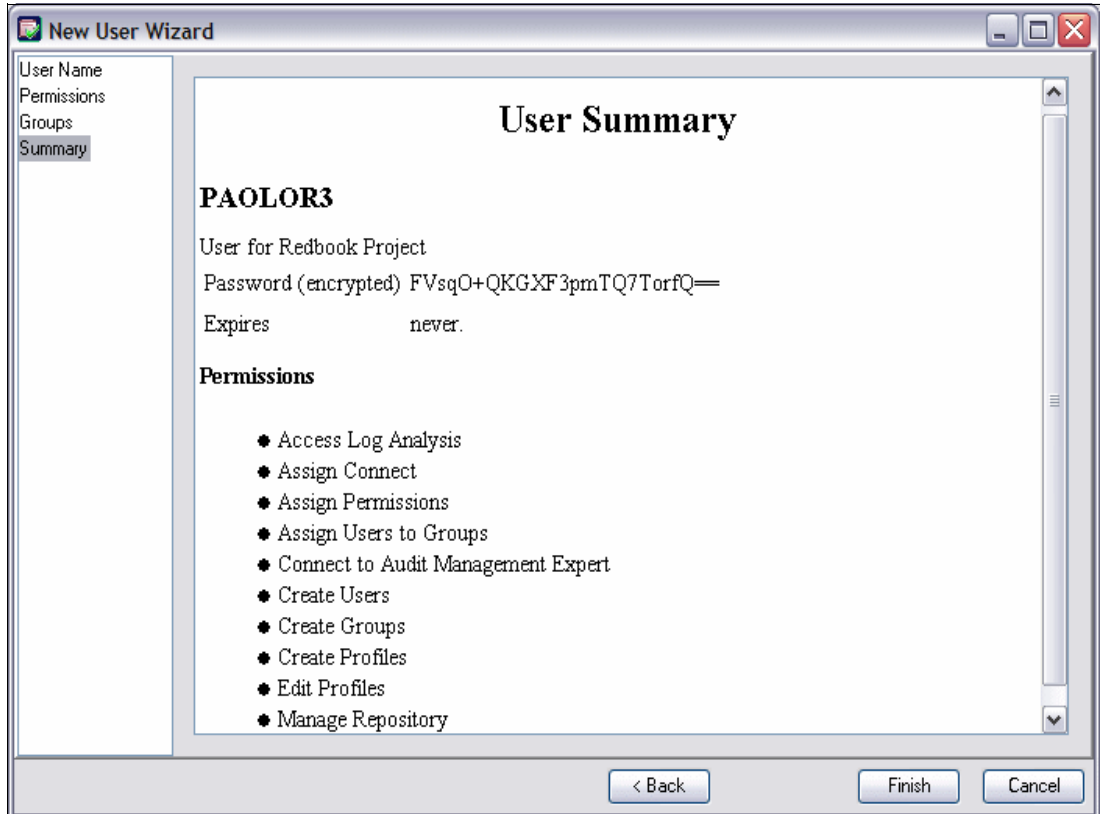


Figure 10-6 User Summary

Click **Finish** to create the new user PAOLOR3 (Figure 10-7). There is one additional line for PAOLOR3, with the check boxes indicating the user's privileges.

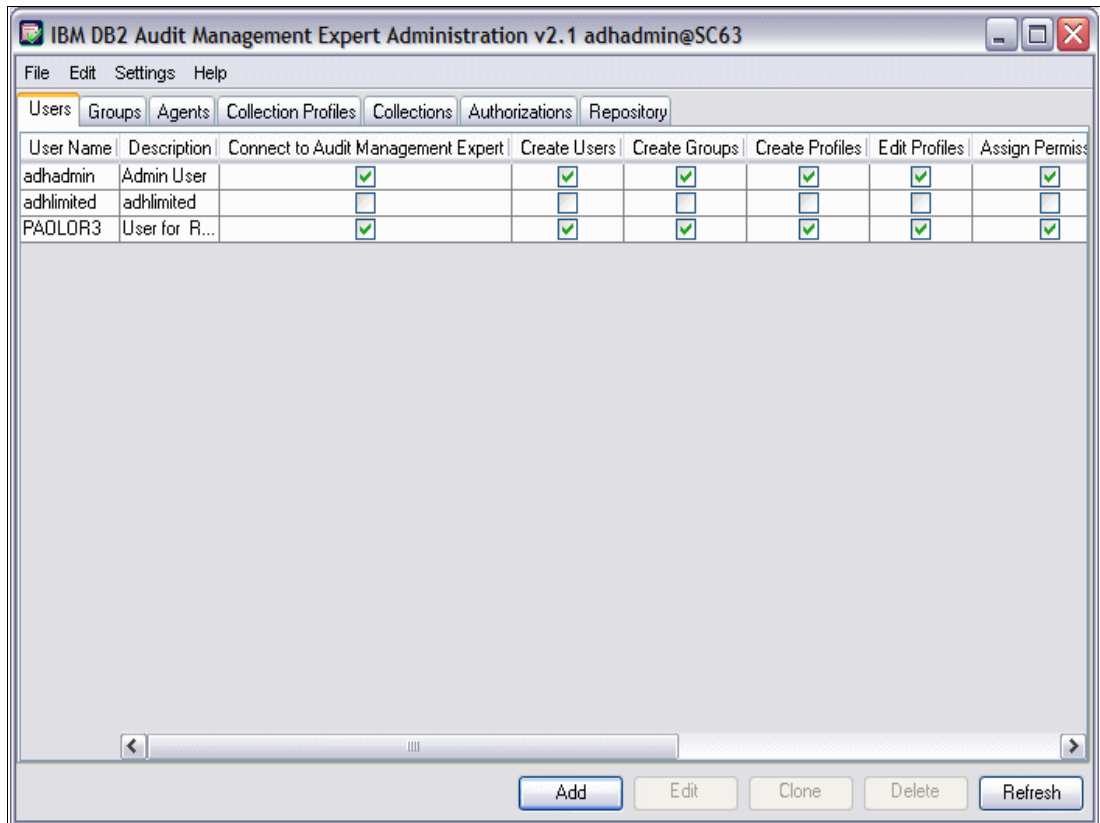


Figure 10-7 User list

After creating a new user, you need to grant the user the right to read the audited data for a certain DB2 subsystem. You need to complete this in the authorizations administration panel. From the administration client, click the **Authorizations** tab.

The **Authorizations** tab displays the user (or group) that has that authorization, the subsystem to which the authorization is applied, whether or not the authorization is active, and the date of the last status change. As shown in Figure 10-8, the user adhadmin has authorization to subsystem SC63:DB9A, and the authorization has been active since 2009-02-13 14:21:23.

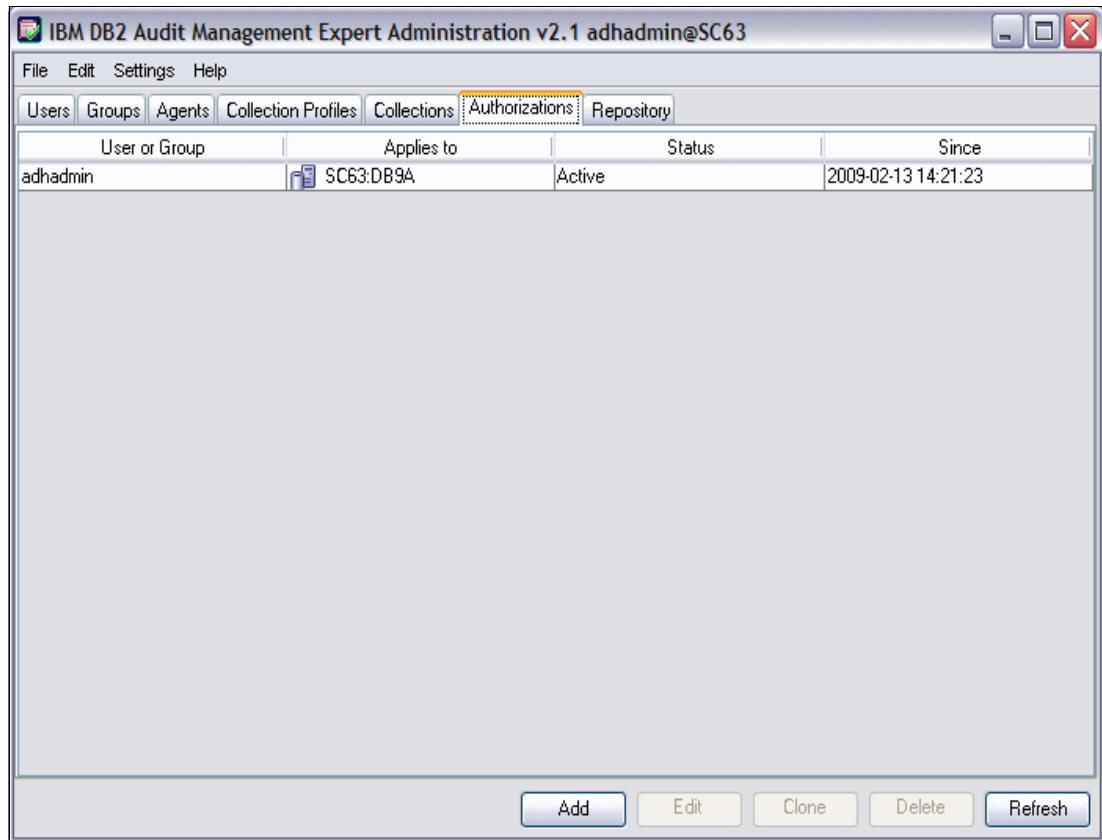


Figure 10-8 Setup user authorizations

To add the new user PAOLOR3, click **Add**. The Authorization Editor appears as shown in Figure 10-9.

In the “User or Group” field, select PAOLOR3.

In the “Status” field, select **Active**.

In the “Applies to” field, select the subsystem SC63:DB9A to which you want to grant the authorizations.

Click **OK** to confirm.

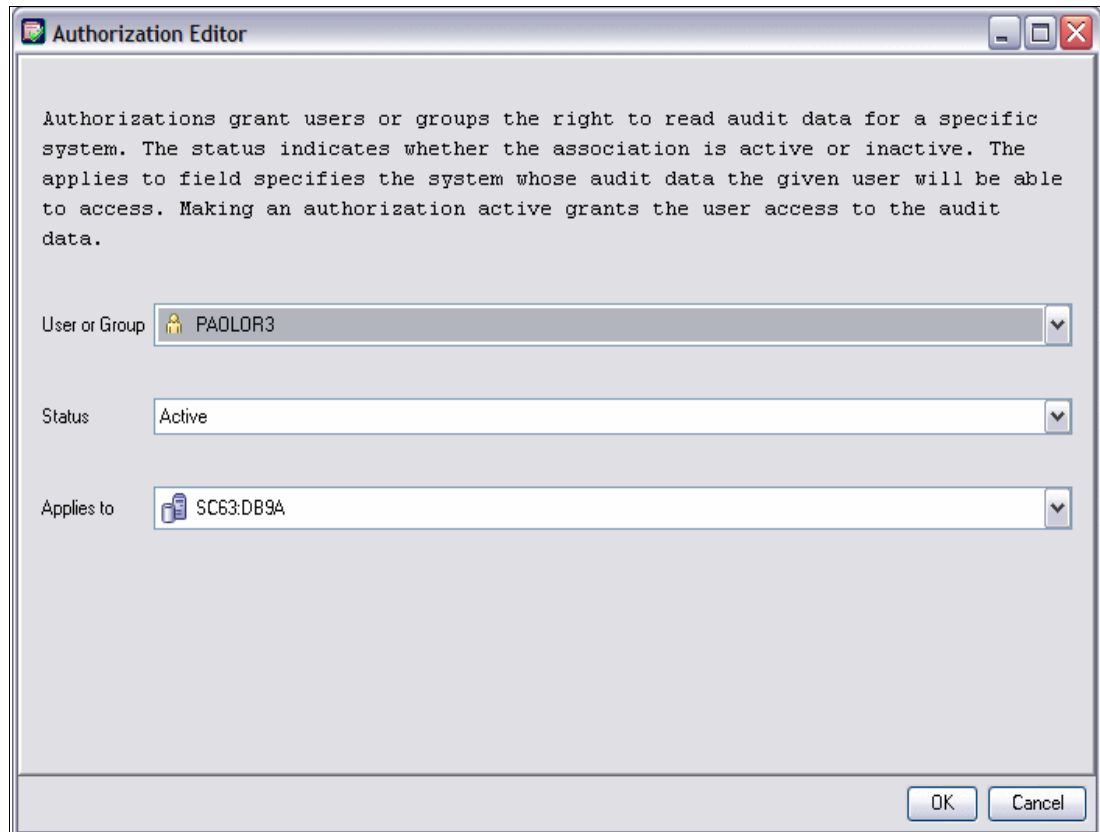


Figure 10-9 Authorization editor

Verify that user PAOLOR3 is now in the authorizations list (Figure 10-10). PAOLOR3 can now access the audited data for DB2 SC63:DB9A.

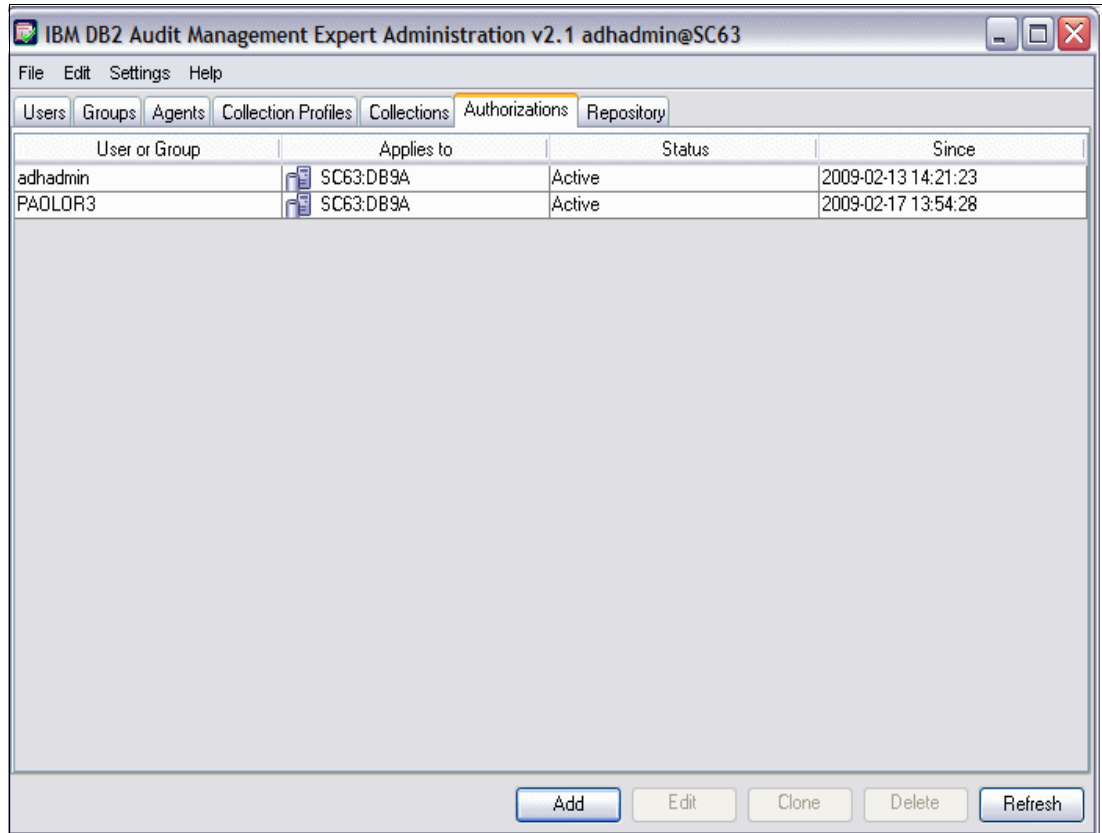


Figure 10-10 Authorizations List

We introduced security control of DB2 Audit Management in 4.1, “DB2 Audit Management Expert for z/OS” on page 92. This scenario shows the powerful functions DB2 Audit Management Expert provides for separation of duties.

With Audit Management Expert, the administration of users and passwords are independent from operating systems and DB2 privileged users. Also, you can assign various privileges to different users and maintain different levels of auditing authorities.

10.2 Reporting User Interface

This section describes the Reporting User Interface of Audit Management Expert. This section contains the following subsections:

- ▶ Introduction to Reporting User Interface
- ▶ Auditing privileged users
- ▶ Finding all authorization failures
- ▶ Finding DDL activity

10.2.1 Introduction to Reporting User Interface

In this section, we give an overview of the Reporting User Interface by highlighting its three main characteristics:

- ▶ Centralization
- ▶ Simplification
- ▶ Automation

Centralization

As systems and data proliferate across the enterprise, centralization is integral to cost reduction, creating easier and more thorough audits, and reducing the risk of being out of compliance. DB2 Audit Management Expert for z/OS has the ability to collect audit data from the DB2 subsystems across your enterprise, then combine and correlate the audit data together, and then display the information in three possible levels of reporting:

- ▶ Level1
- ▶ Level2
- ▶ Level3

Level1 report

A Level1 report displays the summary audit status of all the DB2 subsystems that the auditors have permission to view.

Log on to the reporter. The auditors are located at the Level1 report (Figure 10-11). It displays the overview status of the DB2 subsystems that are currently being audited and are available to the server to which the auditors are connecting. In the reporter, each box represents a DB2 subsystem, with its name and the date range available for reporting displayed within the box. In our example, we have only subsystem SC63: DB9A. In a real case, the auditors might see more, according to their configuration of DB2 Audit Management Expert for z/OS.

Note: There is a minimum requirement of one Audit Management Expert server per sysplex. So if you configure in this way, you can connect to one server and see the audit report of all the DB2 subsystems within the sysplex.

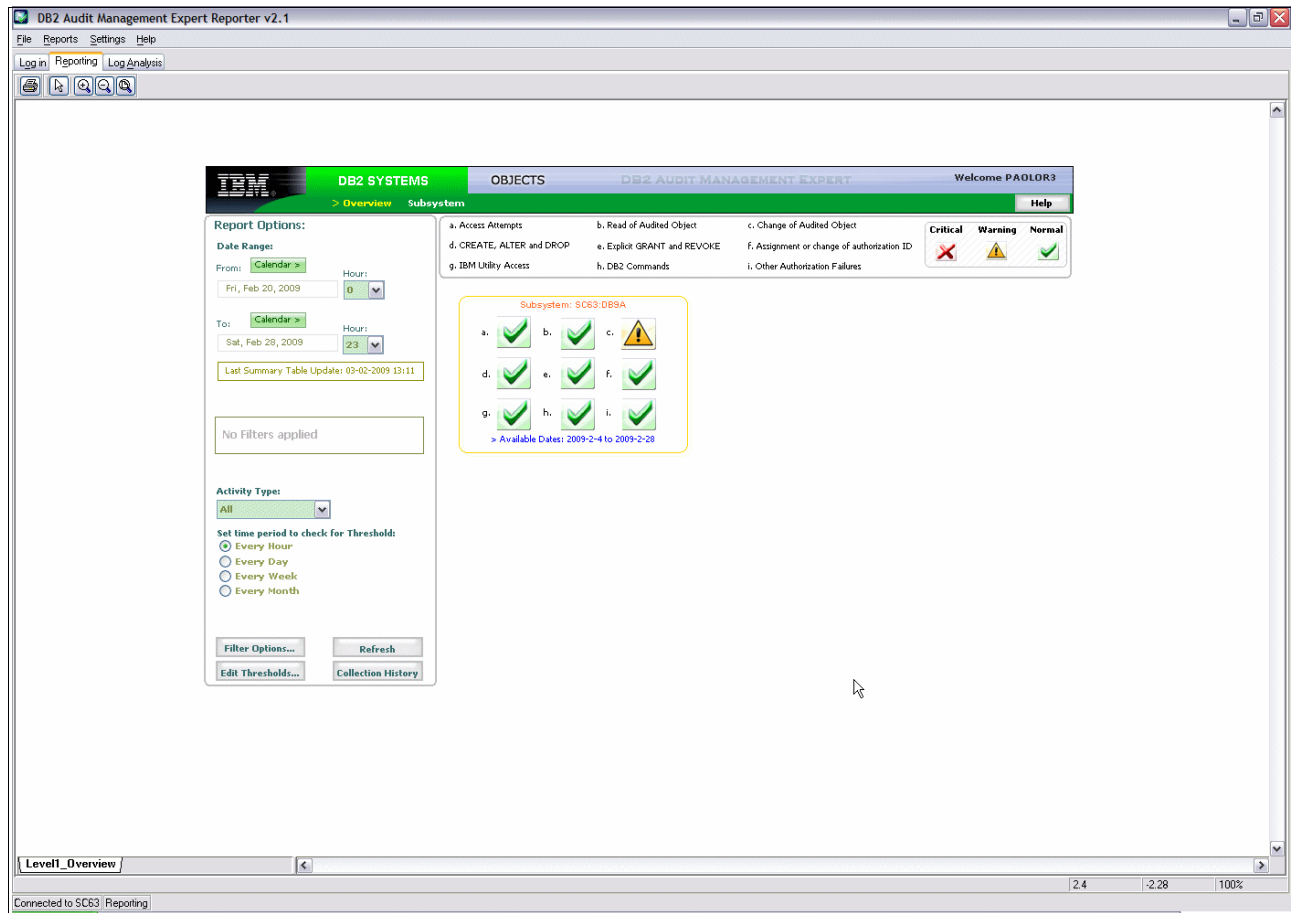


Figure 10-11 Level1 report: Subsystem overview

Level2 report

A Level2 report pertains to a specific DB2 subsystem. The Level2 report (Figure 10-12) displays the summary report for one of the audited subsystems. From this report, the auditors can get an overview status of all the activities of the subsystem being audited. The types of activity are as follows:

- ▶ Access attempts to the subsystem
- ▶ The read of audited objects (SQL SELECT)
- ▶ The change of audited objects (SQL UPDATE, INSERT, DELETE)
- ▶ CREATE, ALTER and DROP operations against an object
- ▶ Explicit GRANT and REVOKE operations
- ▶ Assignment or modification of an authorization ID
- ▶ IBM utilities accesses to an object
- ▶ DB2 commands entered
- ▶ Authorization attempts that are denied due to inadequate privileges

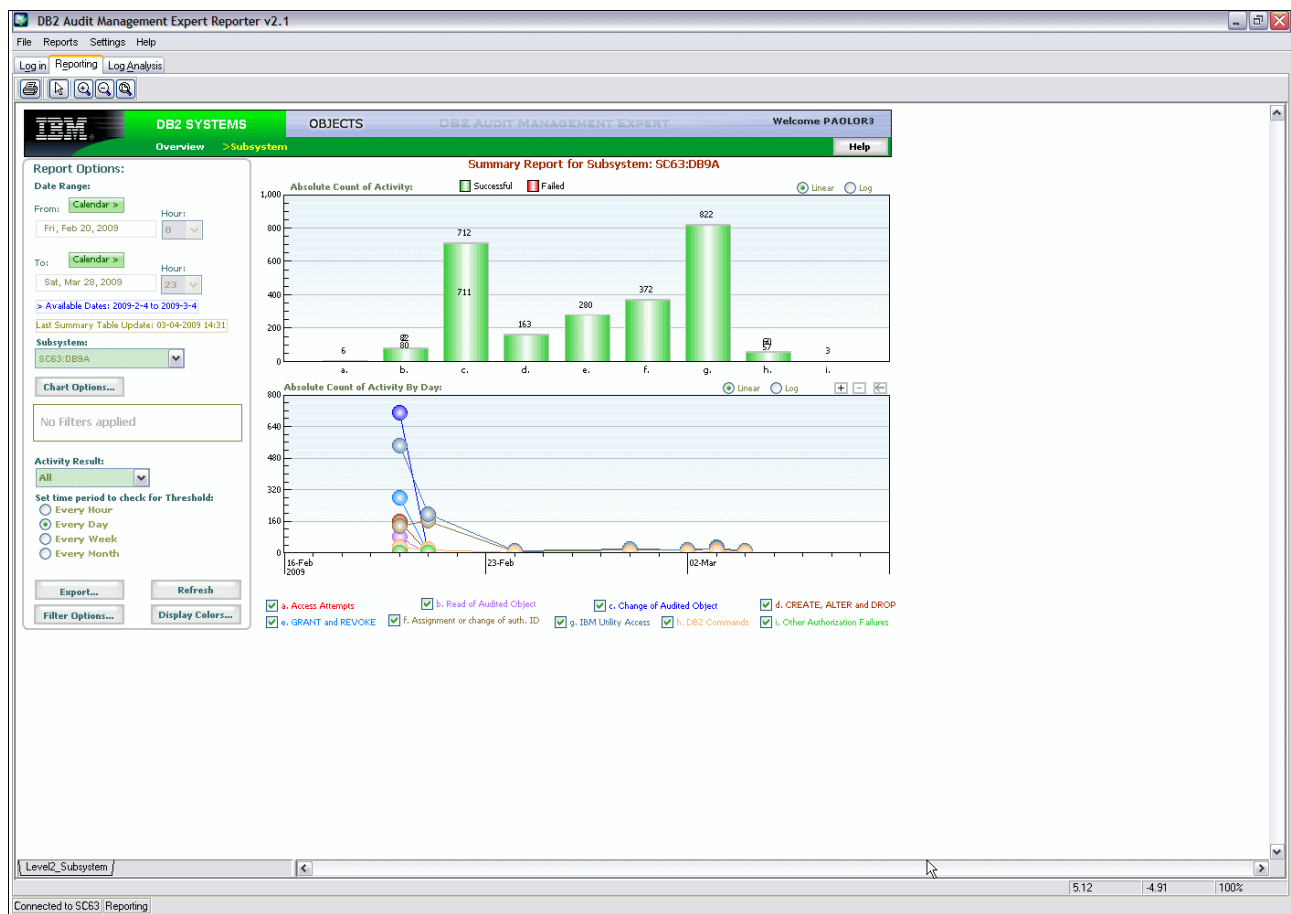


Figure 10-12 Level2 report: Summary report for subsystem

Level3 report

A Level3 report also pertains to a specific DB2 subsystem. The difference from a Level2 report is that it provides more granular information. Auditors can choose either to view the Level2 report from the DB2 subsystem's perspective or from the objects' perspective. Figure 10-13 shows an example of the Level3 report in the subsystem's perspective. It displays the IBM utility access to the audited subsystem. For each type of activity listed in a Level2 report, the Level3 report provides more detailed information. By clicking the histograms in the charts, auditors can get undisputed proof of who did what, where, why, when, and how. In the following scenarios, you will learn more about that.

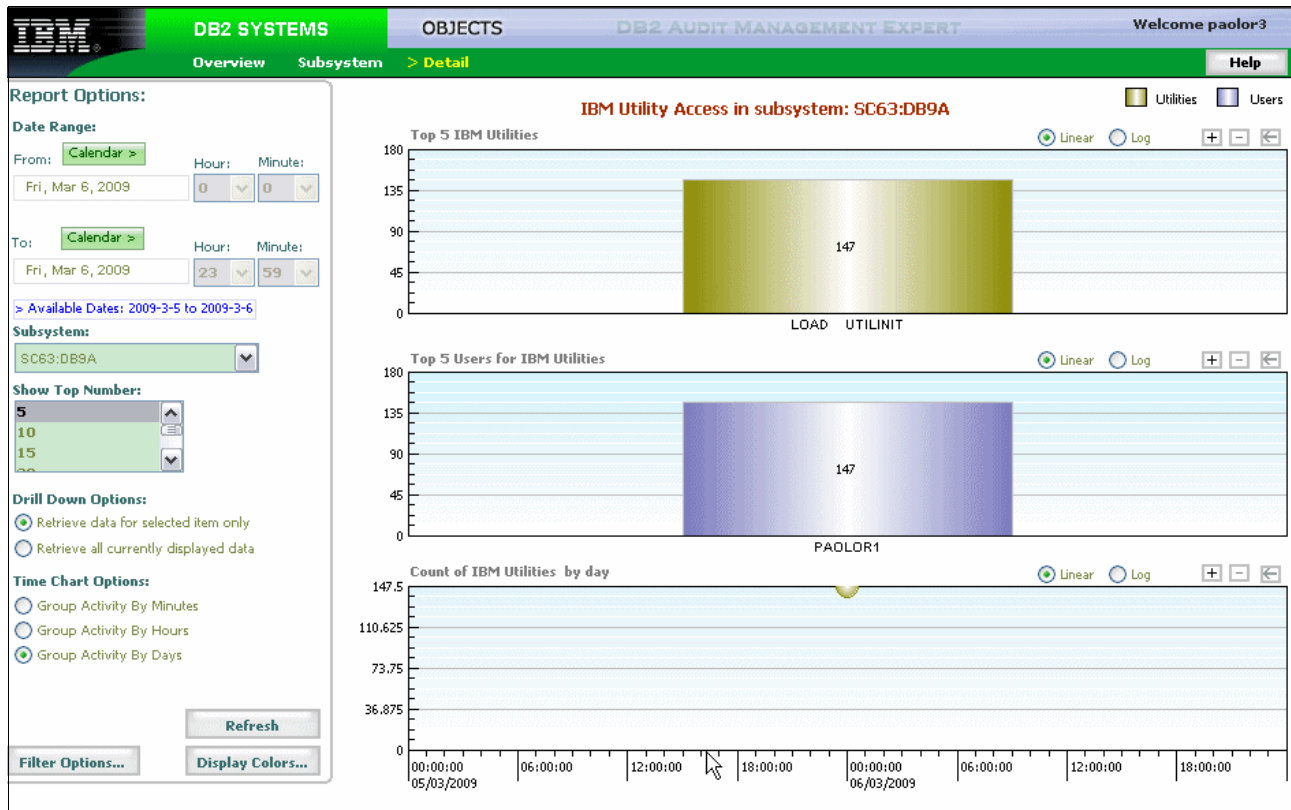


Figure 10-13 Level3 report: Subsystem detail

Figure 10-14 is the Level3 report from the objects' perspective. Auditors can gain a summary status of the activities against the objects under auditing, such as TABLE UPDATE, TABLE CREATE, TABLE DROP, and so forth. By double-clicking the histograms, auditors can get more detailed and comprehensive reports, which enable them to find breaches easily.

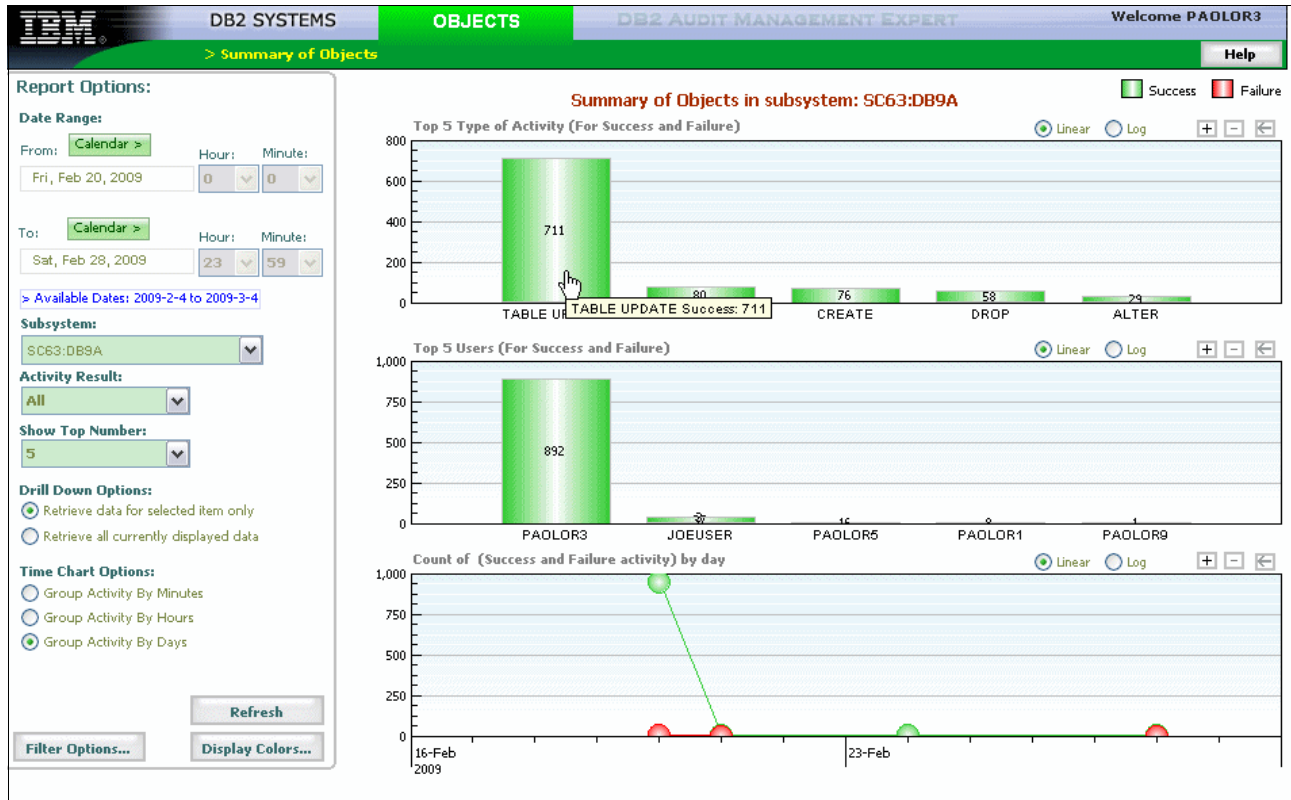


Figure 10-14 Level3 report: Summary of objects in subsystem

Simplification

Audit Management Expert reporter provides auditors with flexible easy-to-use options for examining the data in the repository.

For example, by setting the reporting options, such as date range of report, activity type, filter options, thresholds, and so forth, auditors can create meaningful reports easily. As shown in Figure 10-15, auditors can use the Date Range options to specify the starting and ending date for the report. Also, they can specify the activity type they would like to see in the report, all the activities, or only the successful ones, or only the ones that fail.

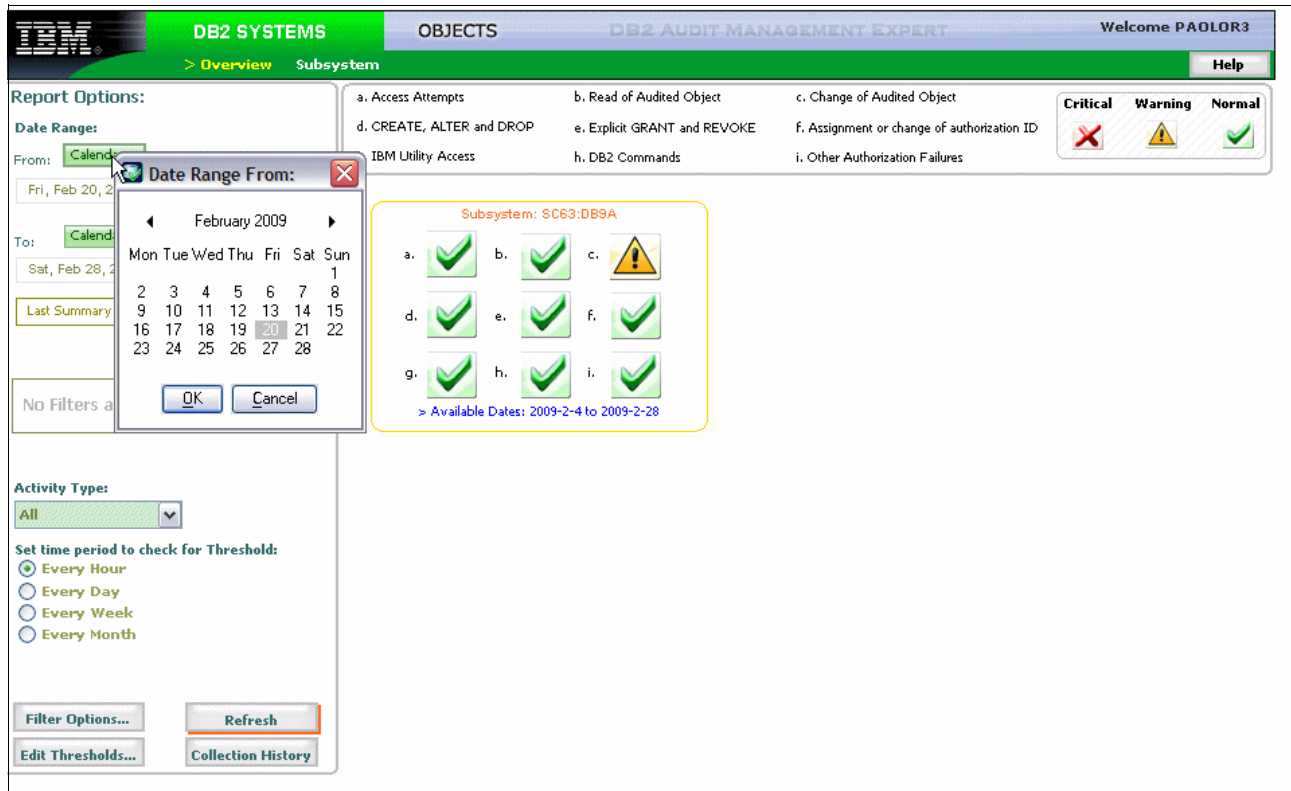


Figure 10-15 Select the date range to report

Another powerful function provided by Audit Management Expert reporter is filtering. The filtering function allows the auditors to filter the results collected in the repository by creating sophisticated inclusion and exclusion conditions.

Based on the report level you are viewing, the filter also has three different levels. For the Level1 and Level2 reports, the available filters are AUTHID and PLANS. You will see an example of how to use it in the following scenarios.

The filter at the Level3 report has more choices, as shown in Figure 10-16.

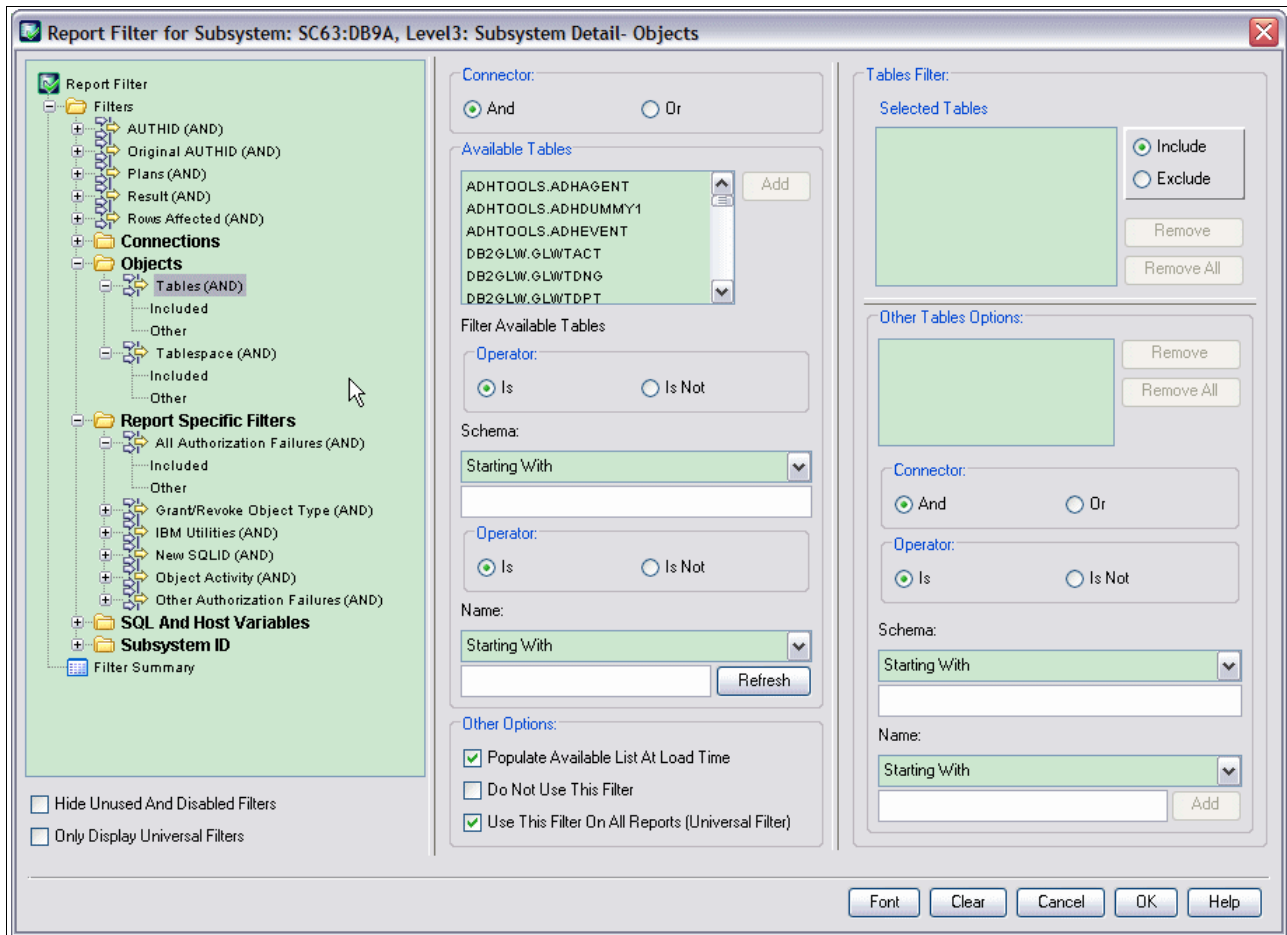


Figure 10-16 Report filter for the subsystem and Level3 report

In Figure 10-16, you can see the comprehensive filter options provided. The filter function enables the auditors to filter out the data they do not need, only displaying the data that they are interested in the report. By using this function, auditors can discover the questionable data easily and quickly.

Automation

The third characteristic that we would like to highlight here is automation. Audit Management Expert can collect audit data, to generate auditing reports automatically. Auditors now have a simple method to gain the information they require to meet compliance. Also, Audit Management Expert reporting has the ability to export the data into CSV files, which can be imported into other applications (such as Microsoft Excel spreadsheets). Auditors can be freed up from spending time generating the complicated reports.

10.2.2 Auditing privileged users

As described in 2.4.2, “Audit versus external security” on page 36, it is crucial for a company to have an auditing process for privileged users. Take database administrators as an example. They do not have the authority to access or update the data in the production system using authorized production processes, such as CICS or IMS, because they are all well protected by RACF. However, database administrators have the DBADM authority, which

enables them to administer the resource in the production system through other mechanisms, which are out of RACF's control. DB2 Audit Management Expert is a good solution for the customer to audit the privileged users' activity, and realize a robust compliance process. In this section, we show two scenarios as examples.

Assume you are an auditor. The company's production DB2 subsystem contains many tables that hold lots of sensitive data, such as employee, department, and project information. The authorized applications that access these tables are protected by RACF, which does not grant privilege to DBAs to execute those applications. However, because DBAs have other ways to access the data, the company's auditing process requires tracking and recording the DBA's activity (or the activity of any privileged user) on a daily basis.

You will see how DB2 Audit Management Expert helps you answer the following questions:

- ▶ Was there any access of type update to the sensitive data outside the authorized production process?
- ▶ If yes, who did it?
- ▶ When and how?

Privileged user updates sensitive data

The first scenario shows DB2 Audit Management Expert for z/OS capability of tracking and recording a privileged user's updates to the production DB2 subsystem.

Log on the Reporter UI, and set the date range. You can get an overview auditing status of the DB2 subsystem for the day of Feb. 20, 2009 (Figure 10-17).

You can change the date range and time range according to your auditing requirements.

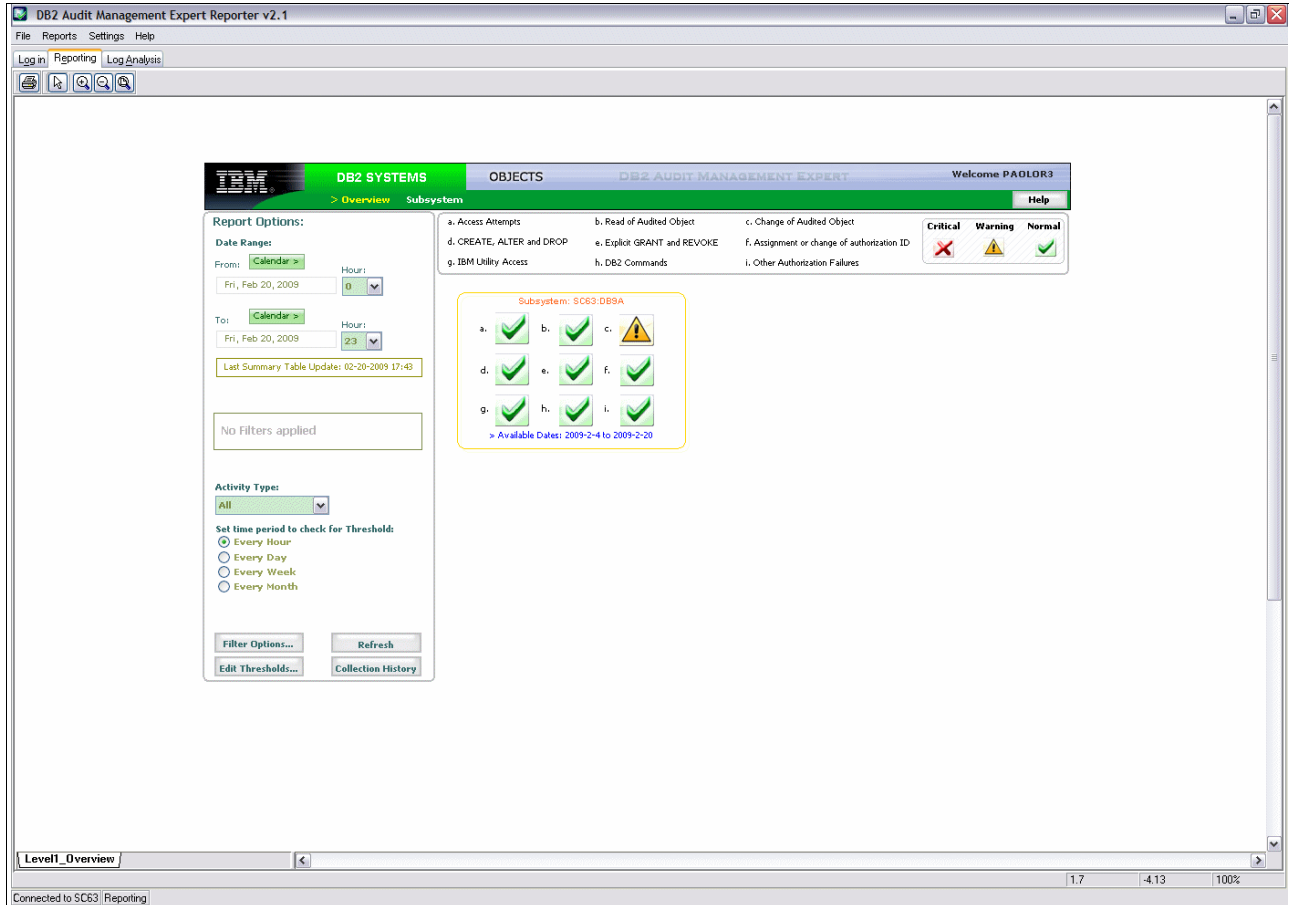


Figure 10-17 Overview of the audited DB2 subsystem

DB2 Audit Management Expert provides a filter function in the reporter, which enables you to filter out the data in which you have no interest, and focus on the ones you care about. Click **Filter Options**. You will see the “Report Filter” window shown in Figure 10-18.

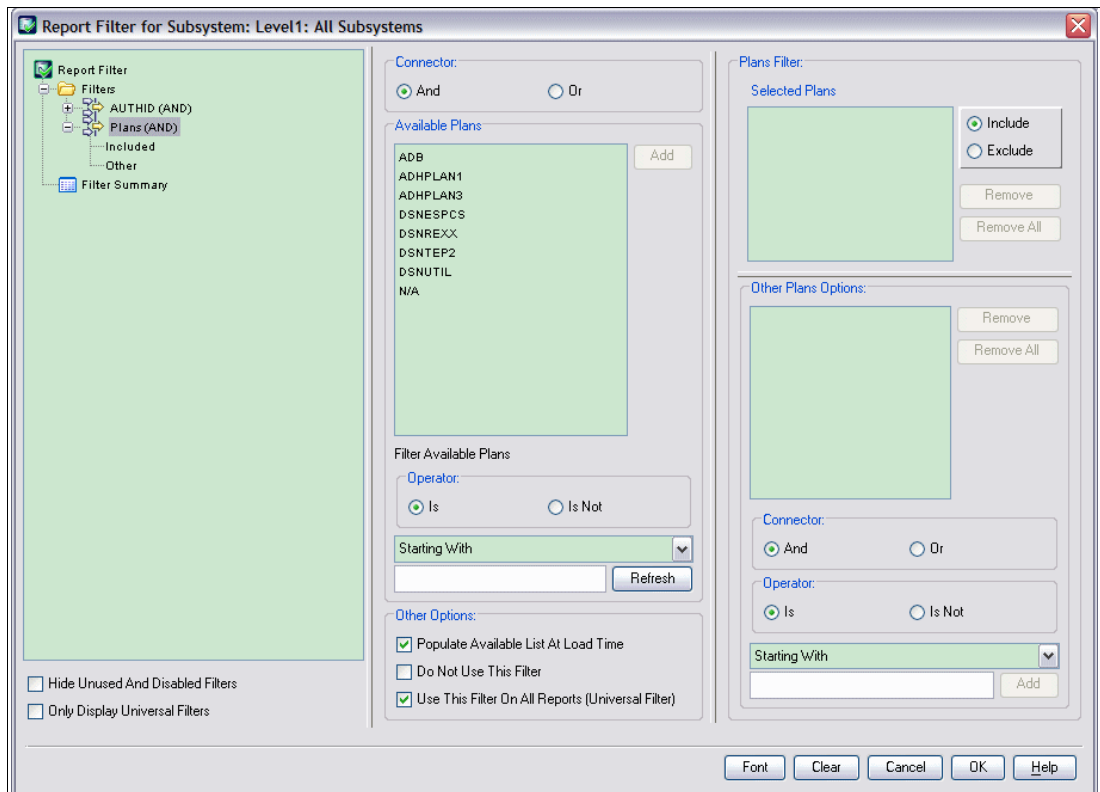


Figure 10-18 Report filter for subsystem

In this window, you can see the filter criteria provided by DB2 Audit Management Expert. In the Level1 report, you can choose to filter by AUTHID or by plan. In this scenario, we want to audit the activities of the privileged users. Is there any update to the sensitive data that may cause security violation? The most effective way to determine this is to filter by plan, excluding all the plans that are authorized by the production process, so that the report will include only the questionable data.

In our example, the authorized plan is DSNREXX. So we excluded the executions of this plan, as shown in Figure 10-19. In a real production system, it might be several others. The filter function of Audit Management Expert makes your auditing more efficient.

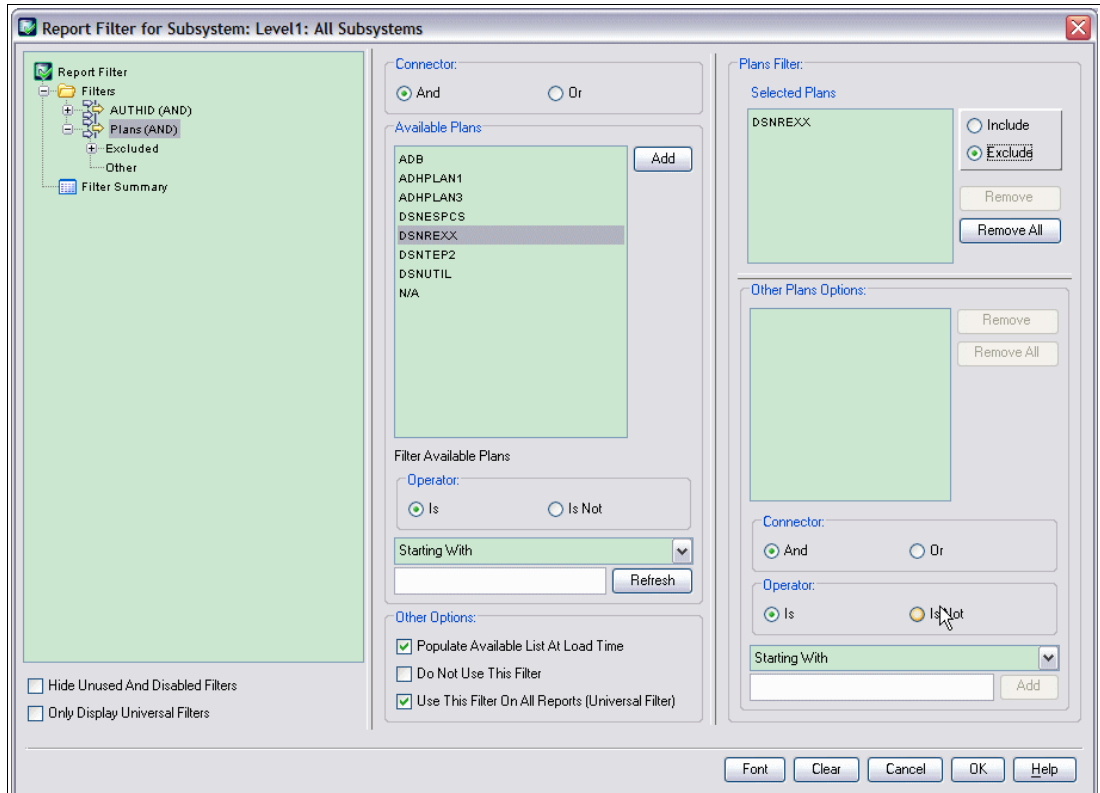


Figure 10-19 Exclude plans in Report Filter window

With the filter criteria defined, return to the main window and refresh the report. As shown in Figure 10-20, now you can see there is a note in red indicating which filters are currently applied. Also, you can find in the subsystem overview report that type 'c' changed from *warning* to *normal*. That is because we have filtered out activities that are normal and compliant.

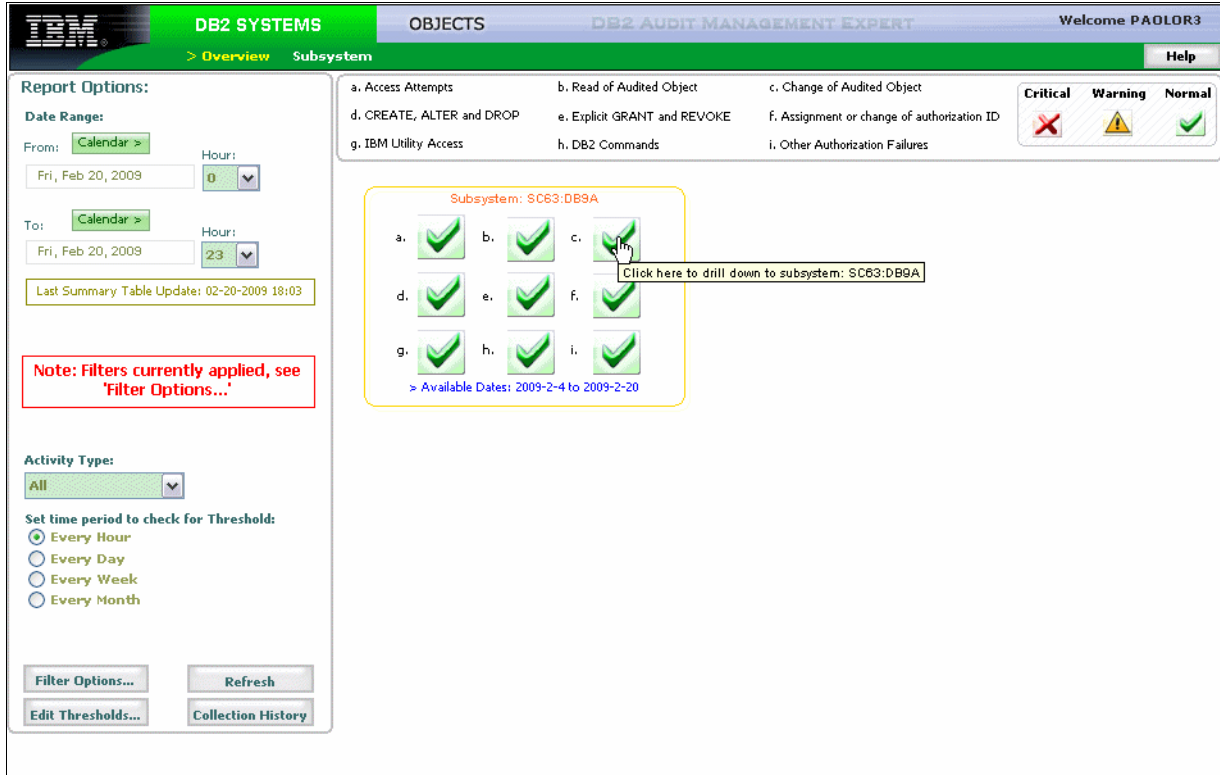


Figure 10-20 Overview of the audited subsystem with filter applied

Click any type of audited activity for the subsystem. You can drill down to see a more detailed report.

In Figure 10-21, you can find a summary audit status of the DB2 subsystem. On the first chart, it displays the absolute count of each activity. Here 'a' to 'i' each represents a type of activity. The descriptions are at the bottom of the window. There are 7 changes in total made on the audited objects. These changes were done outside the authorized production process because we have filtered out the authorized plans. Are there any violations? Who did it? What kind of changes have been made? Double-click the histogram to find the answers to these questions.

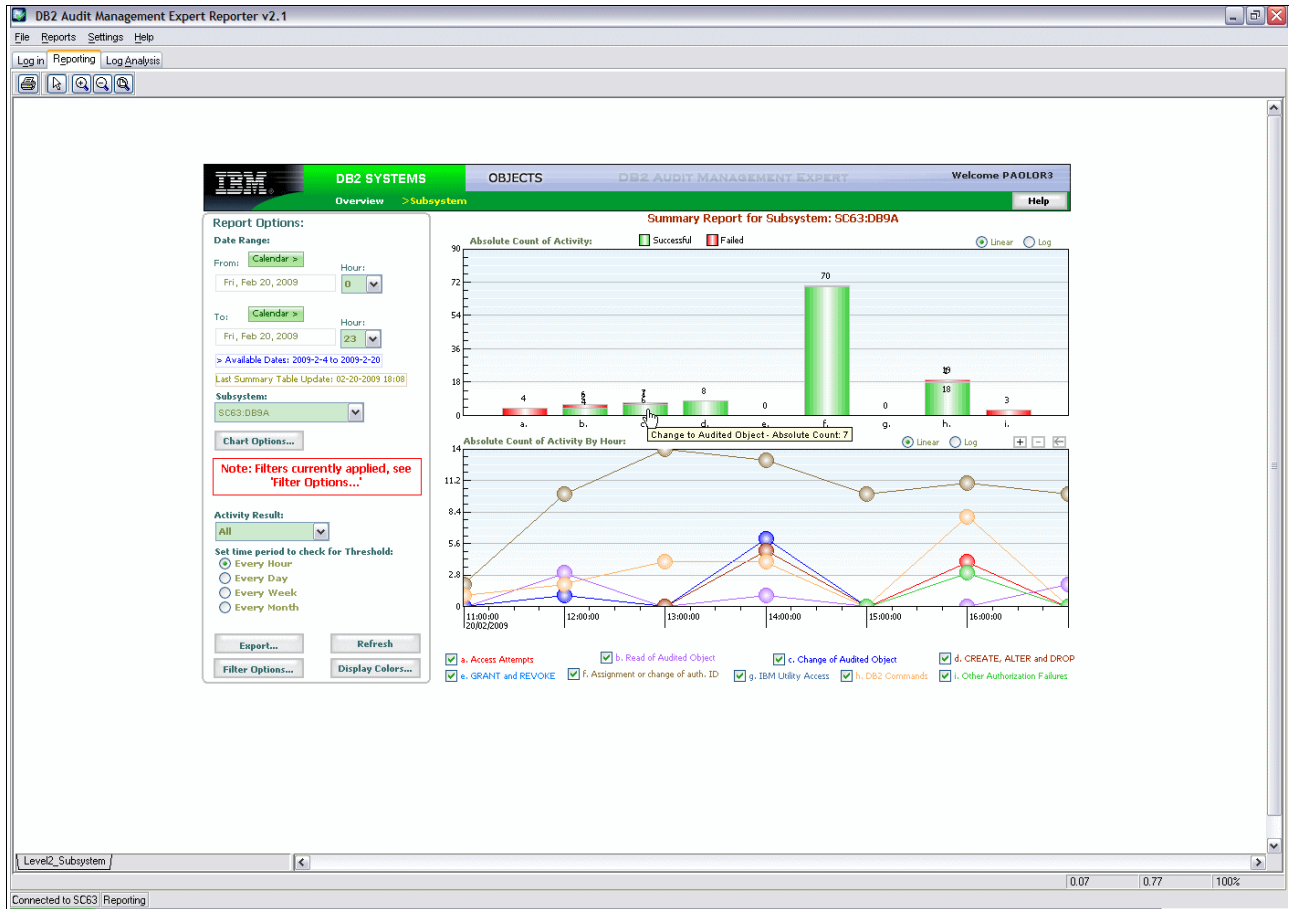


Figure 10-21 Summary report for the audited DB2 subsystem

The window in Figure 10-22 on page 235 gives us a comprehensive report on the change of audited objects for the DB2 subsystem. From the Report Option panel you can see, as you drill down to the detailed report of a type of audited activity, that the time granularity is narrowed down from hour to minute. You can select to view the audit report of a smaller interval of time. DB2 Audit Management Expert provides different granularity for different levels of report, which is flexible to the auditors.

On the top of the charts, you can still find the red note that reminds you that filters are being used. Once filters are applied, Audit Management Expert displays this note on every report that has the filters on. This user-friendly feature avoids missing any of the questionable data.

The report shows both the successful and failed changes of the audited objects from different point of views. The first chart at the top targets the tables under auditing, while the second chart shows the activity related to the audited users. The third chart is a time-based report, which gives you an overview of changes, when and how many changes happened.

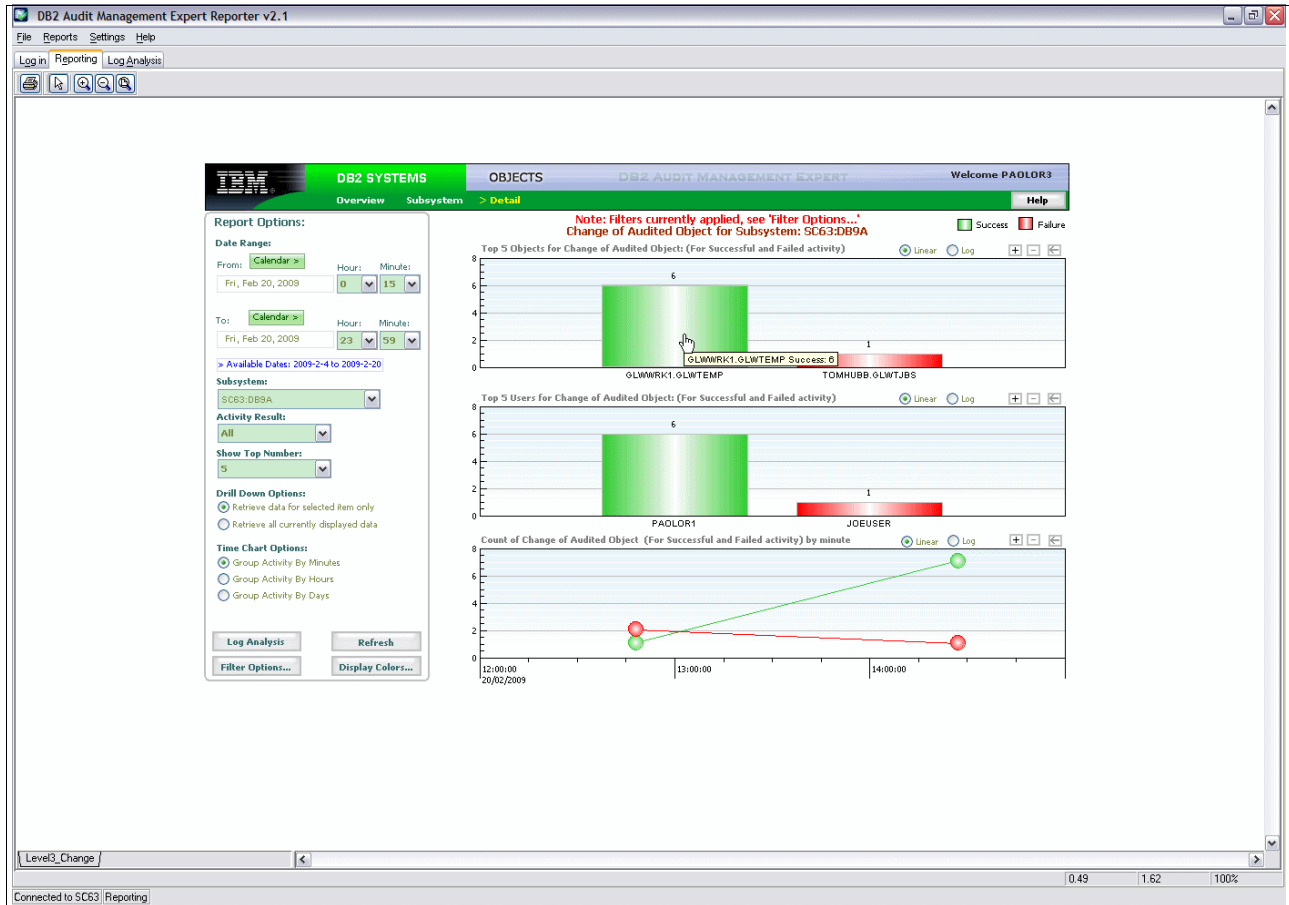


Figure 10-22 Change of audited object for the DB2 subsystem

From the report you can find that there are six successful changes to the employee table, which are questionable. Audit Management Expert provides more detailed reports that can help you determine if there are any breaches.

Double-click the histogram of successful changes to the employee table. You will see the comprehensive Level3 report shown in Figure 10-23.

R...	TIME	RETURN...	SCHEMA	NAME	IFICODE	CONTEXT_TYPE	TYPE	AUTHORIZATION_ID	ORIGINAL_OP_ID	STATEMENT_TXT	PLAN	CONNEC...
1	2009-02-20 14:27:2...	SUCCESS	GLMWRK1	GLWTEMP	00143	TABLE UPDATE	TABLE/VIEW	PAOLR1	PAOLR1	UPDATE GLMWRK1.GLWTEMP...	DSNESPCS	TSD
2	2009-02-20 14:27:2...	SUCCESS	GLMWRK1	GLWTEMP	00143	TABLE UPDATE	TABLE/VIEW	PAOLR1	PAOLR1	UPDATE GLMWRK1.GLWTEMP...	DSNESPCS	TSD
3	2009-02-20 14:27:2...	SUCCESS	GLMWRK1	GLWTEMP	00143	TABLE UPDATE	TABLE/VIEW	PAOLR1	PAOLR1	UPDATE GLMWRK1.GLWTEMP...	DSNESPCS	TSD
4	2009-02-20 14:27:2...	SUCCESS	GLMWRK1	GLWTEMP	00143	TABLE UPDATE	TABLE/VIEW	PAOLR1	PAOLR1	UPDATE GLMWRK1.GLWTEMP...	DSNESPCS	TSD
5	2009-02-20 14:27:2...	SUCCESS	GLMWRK1	GLWTEMP	00143	TABLE UPDATE	TABLE/VIEW	PAOLR1	PAOLR1	UPDATE GLMWRK1.GLWTEMP...	DSNESPCS	TSD
6	2009-02-20 14:27:2...	SUCCESS	GLMWRK1	GLWTEMP	00143	TABLE UPDATE	TABLE/VIEW	PAOLR1	PAOLR1	UPDATE GLMWRK1.GLWTEMP...	DSNESPCS	TSD

Figure 10-23 Level3 report for the successful change of audited object

You can find from the report that user PAOLR1, who has the role of DBA, updated the employee table six times successfully using SPUFI. By clicking one row of the STATEMENT_TXT column, you can get further information. Take the first record, for example.

It shows that the DBA updated one employee's salary to 1000000 (Figure 10-24). This is evidence of a security violation. The privileged user performed updates to the company's sensitive data that are not acceptable by the regulatory compliance of the company.

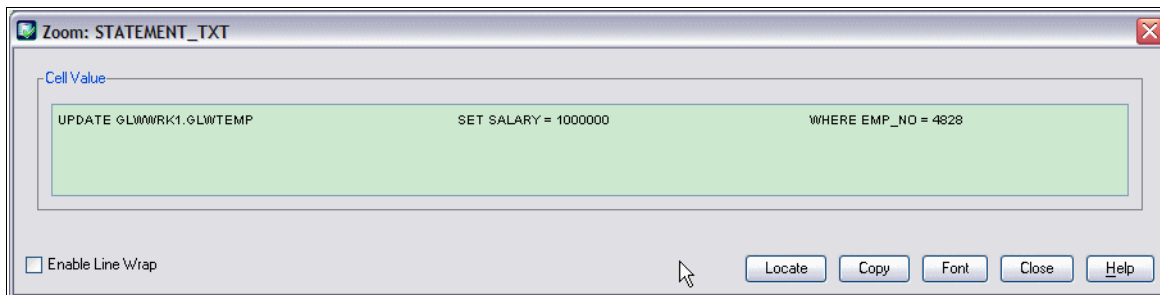


Figure 10-24 Statement executed against the audited object

You will find a user-friendly report saving feature from DB2 Audit Management Expert. As shown in Figure 10-23 on page 235, there are several buttons at the left corner of the window. If you click **Copy**, the report is copied to your clipboard. If you click **Export**, the report is saved to a CSV file. Because Microsoft Excel is installed, it will be associated by default. Go back to the report shown in Figure 10-22 on page 235 to save that report for later retrieval (Figure 10-25).

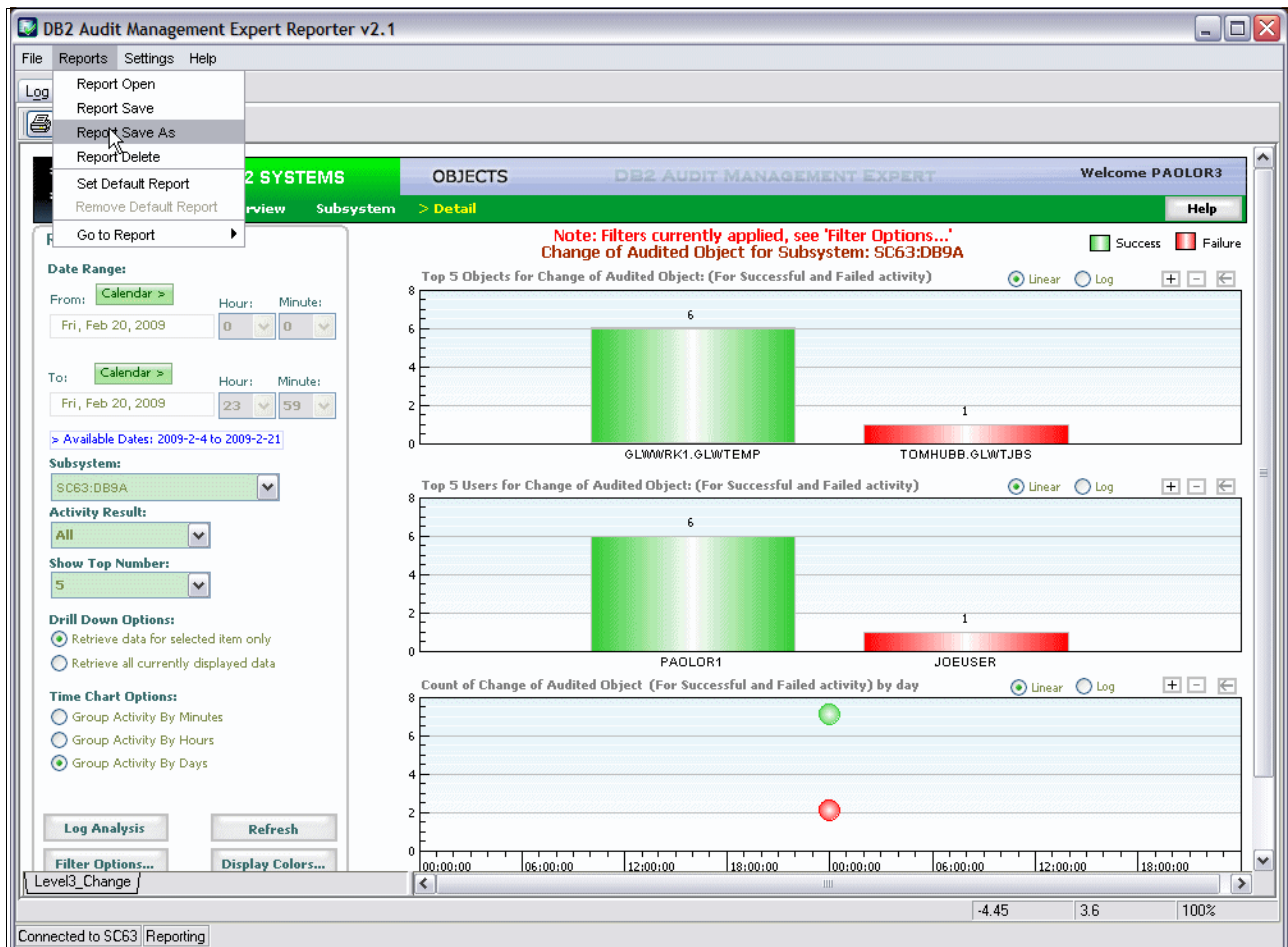


Figure 10-25 Save report

You will receive confirmation as shown in Figure 10-26.

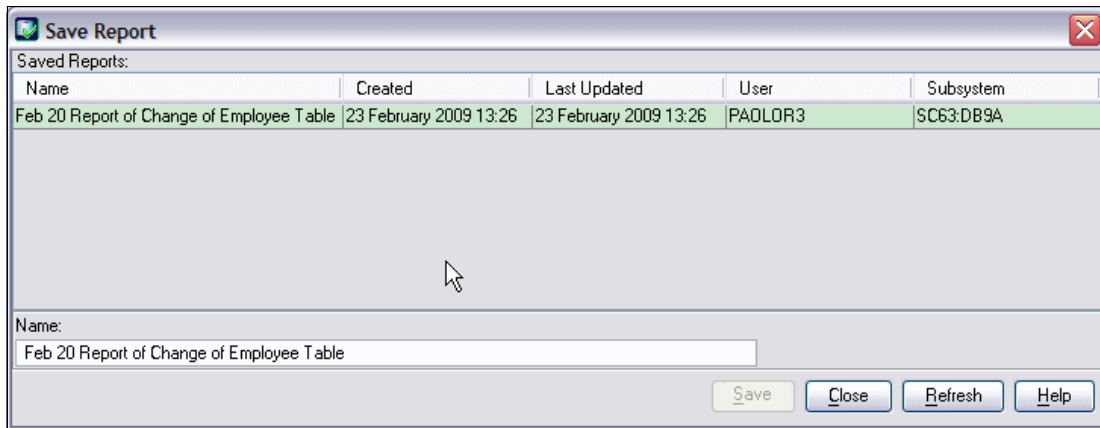


Figure 10-26 Save report for later retrieval

Privileged user accesses sensitive data

In this section, we show how DB2 Audit Management Expert helps you track and record a privileged user's access to your company's sensitive data. Log on to Reporter UI, set the date range for auditing, refresh the reporter, and you will see the overview auditing status for your DB2 subsystem (Figure 10-27).

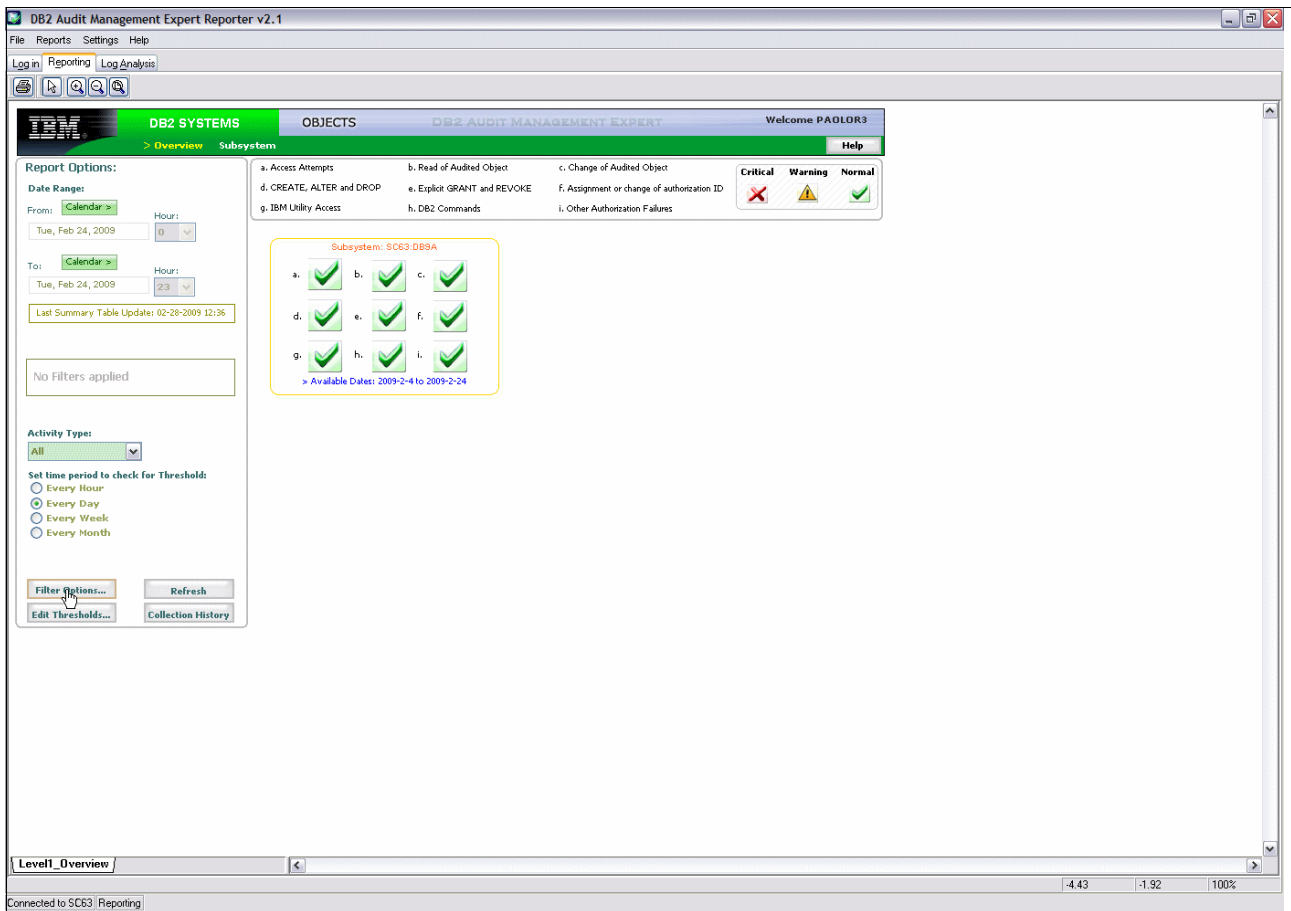


Figure 10-27 Overview of the audited DB2 subsystem

Click **Filter Options** to filter out the production authorized plan. In our scenario, it is DSNREXX. See Figure 10-28.

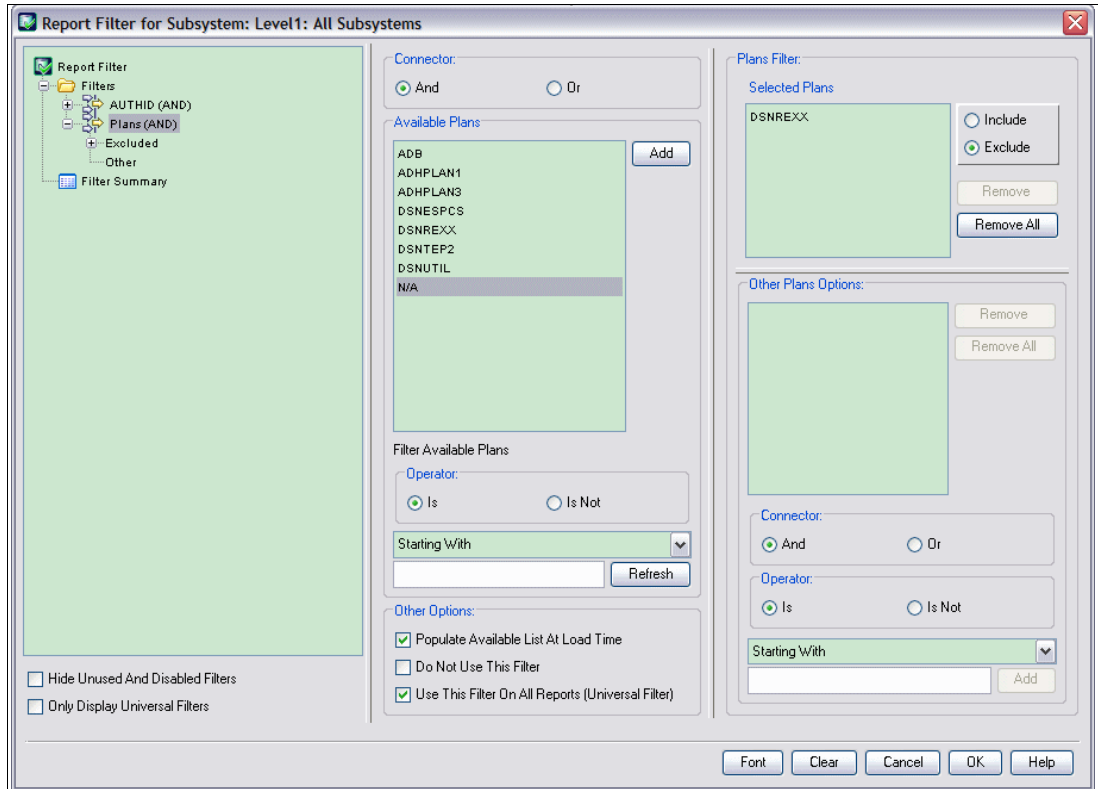


Figure 10-28 Exclude the production authorized plan in the filter panel

With the filter defined, go back to the DB2 Subsystem Overview panel of the Reporter. Refresh. See Figure 10-29.

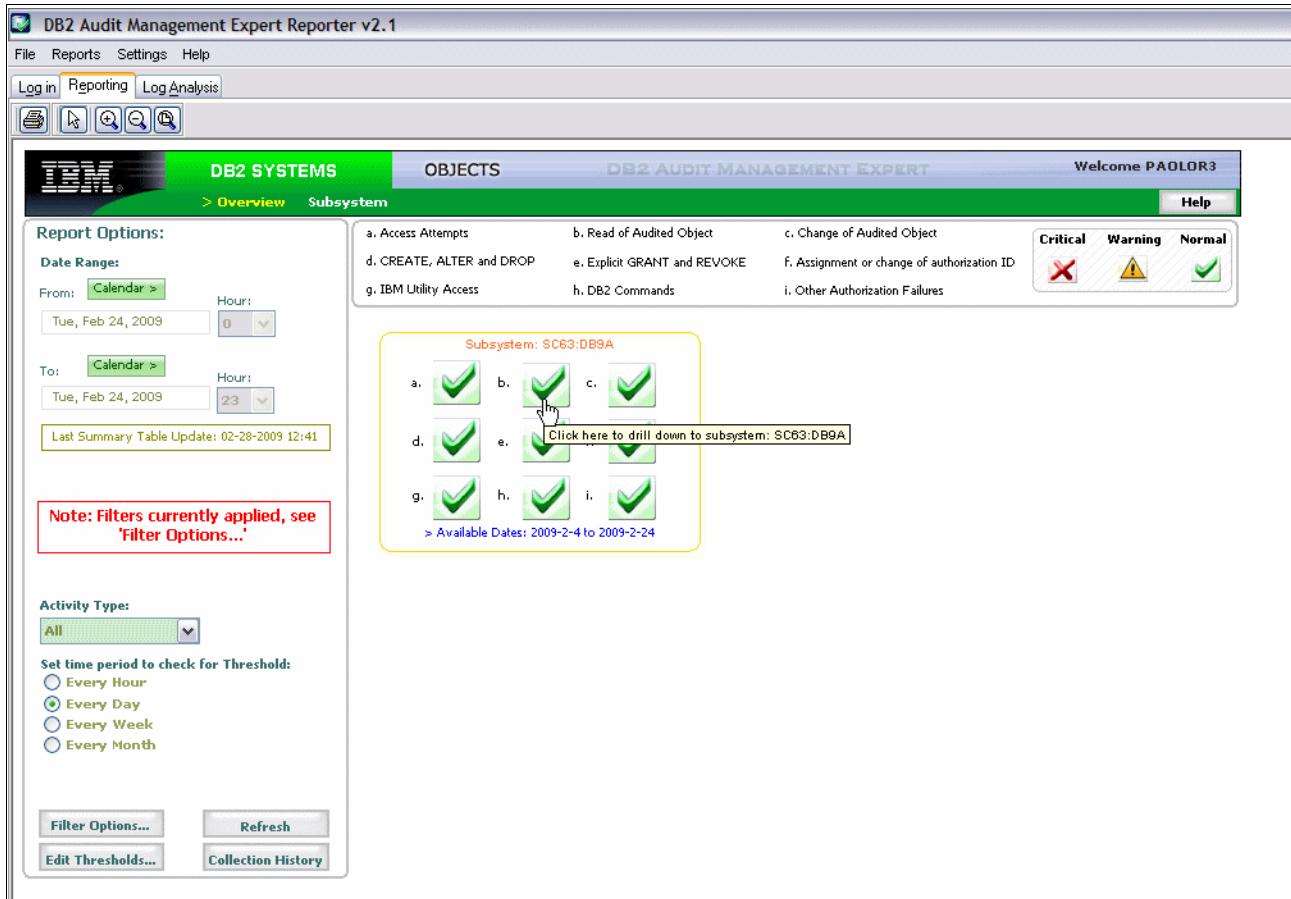


Figure 10-29 Overview of audited subsystem with filter applied

Go deeper to view the summary report of the audited DB2 subsystem, as shown in Figure 10-30.

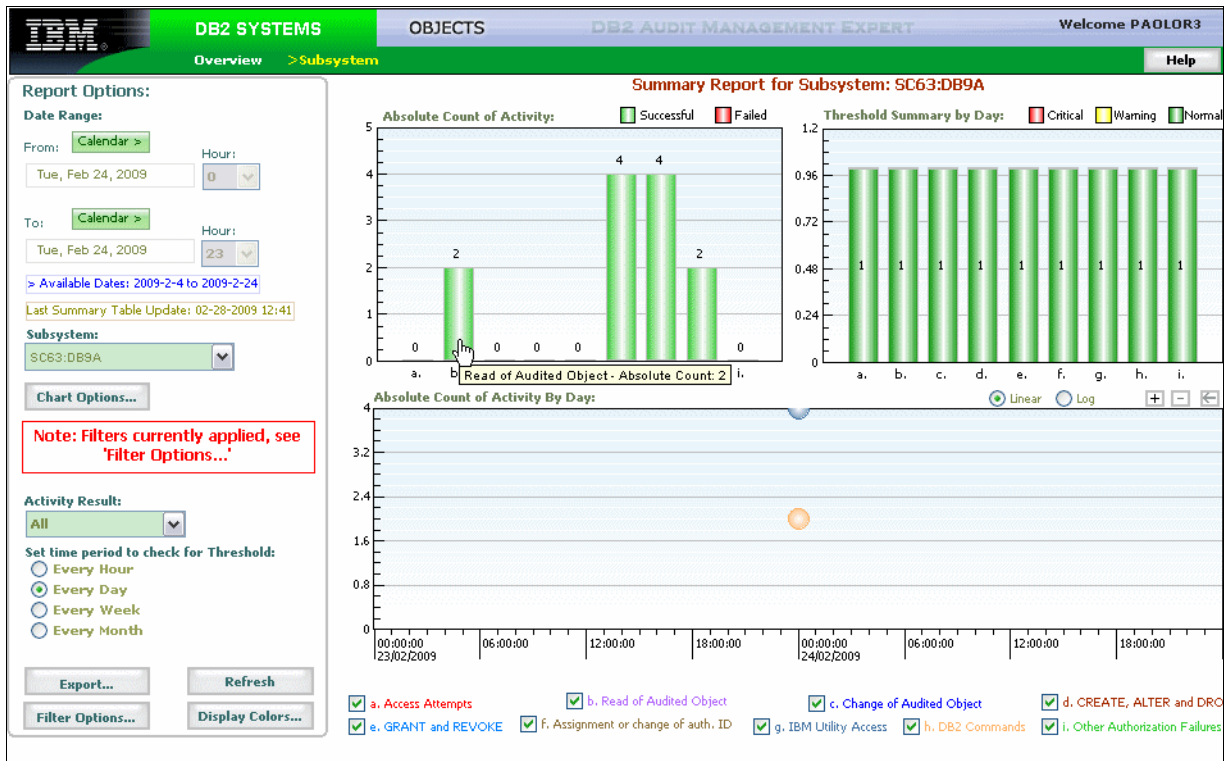


Figure 10-30 Summary report of the audited subsystem

We want to see if there is any access to the company's sensitive data by the privileged user outside of the production-authorized applications. We chose the 'Read of Audited Object Report' shown in Figure 10-31.

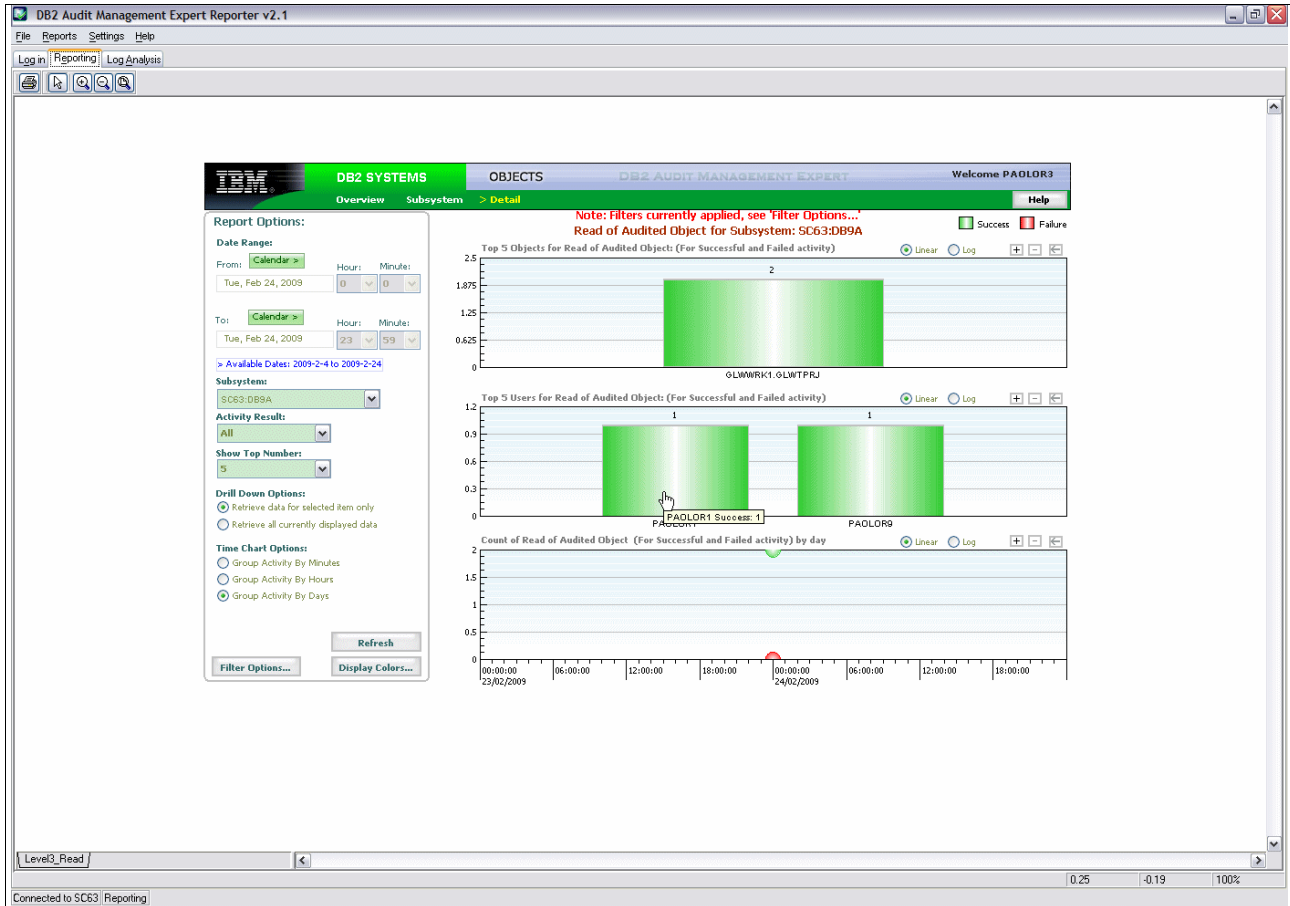


Figure 10-31 Read of audited objects for the subsystem

From this report we find that the DBA, PAOLOR1, has one successful read, which is questionable. Double-click the histogram to get the detailed information in the Level3 report shown in Figure 10-32.

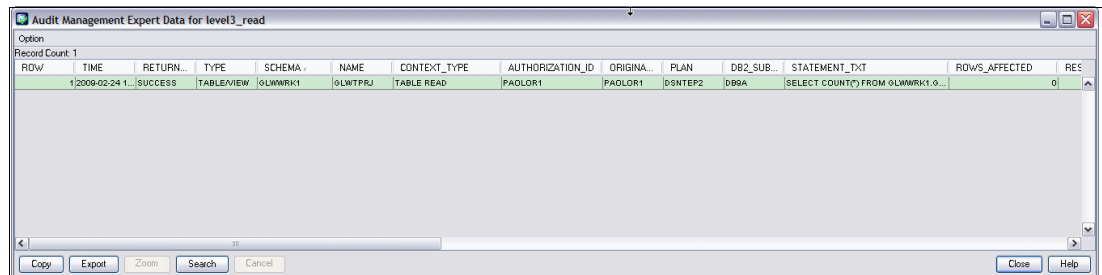


Figure 10-32 Level3 report of audited object read

This report is comprehensive. We find that the DBA accessed the table GLWWRK1.GLWTPRJ using plan DSNTPE2, what time the access was executed, how many rows of the table affected, and so forth.

There is one point we would like to emphasize. Compared with DB2 audit trace, DB2 Audit Management Expert provides more powerful functions on auditing table access. Using DB2 audit trace, if an agent or transaction accesses a table more than once in a single unit of recovery, you can only see the first access. However, with DB2 Audit Management Expert, you are able to find every access to the audited objects in a single unit of recovery.

From these two scenarios, you can see that DB2 Audit Management Expert provides a deep level of auditing activities for DB2. It can trace, record, and report the activities of privileged users at a granular level, insuring that sensitive data is protected. Auditors can easily find breaches of the privileged users. In addition, it provides a filter function that enables users to separate the interesting data from a mass of audited data, helping better manage the audited data and generate meaningful auditing reports efficiently.

10.2.3 Finding all authorization failures

These reports show how to find all authorization failure activity that has occurred on DB2 subsystems that are being monitored by Audit Management Expert.

Note: To generate this report, the target tables must exist in the subsystem and an audit Management Expert agent must be active for that particular subsystem.

Expected results

Audit Management Expert produces a comprehensive report of unauthorized users activity for auditors. It helps auditors to perform the following tasks:

- ▶ Determine unauthorized users failure trying to access a DB2 subsystem.
- ▶ Determine unauthorized users failure trying to set an unauthorized SQLID.
- ▶ Determine unauthorized users failure trying to select a monitored table.

Report 1: Determine unauthorized users failure trying to access a DB2 subsystem

The DB2 SYSTEMS tab provides an overview of the DB2 subsystems currently being monitored by Audit Management Expert. See Figure 10-33.

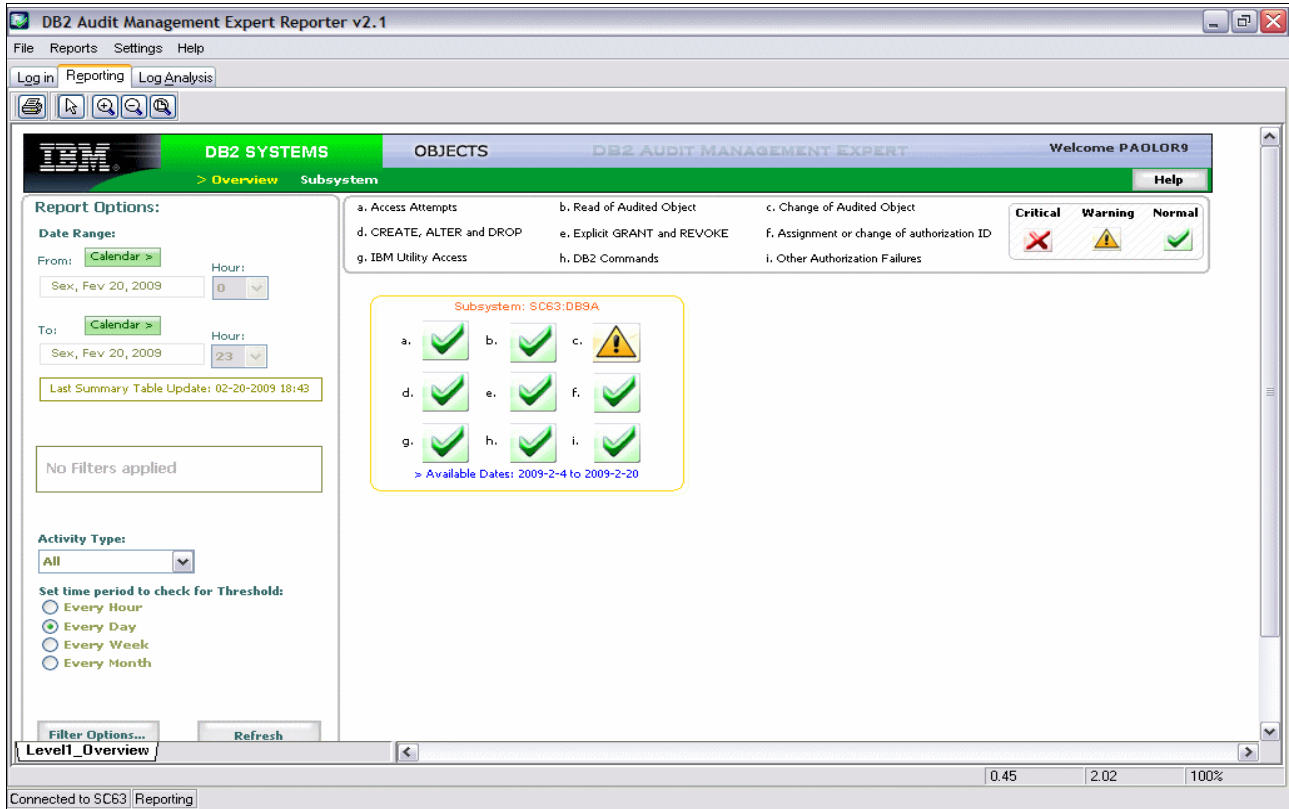


Figure 10-33 Overview of monitored DB2 subsystem (DB9A) on system SC63

On DB2 SYSTEMS tab we select activity types **only failures** (Figure 10-34). We also provide the time interval to be reported. Click **Refresh**.

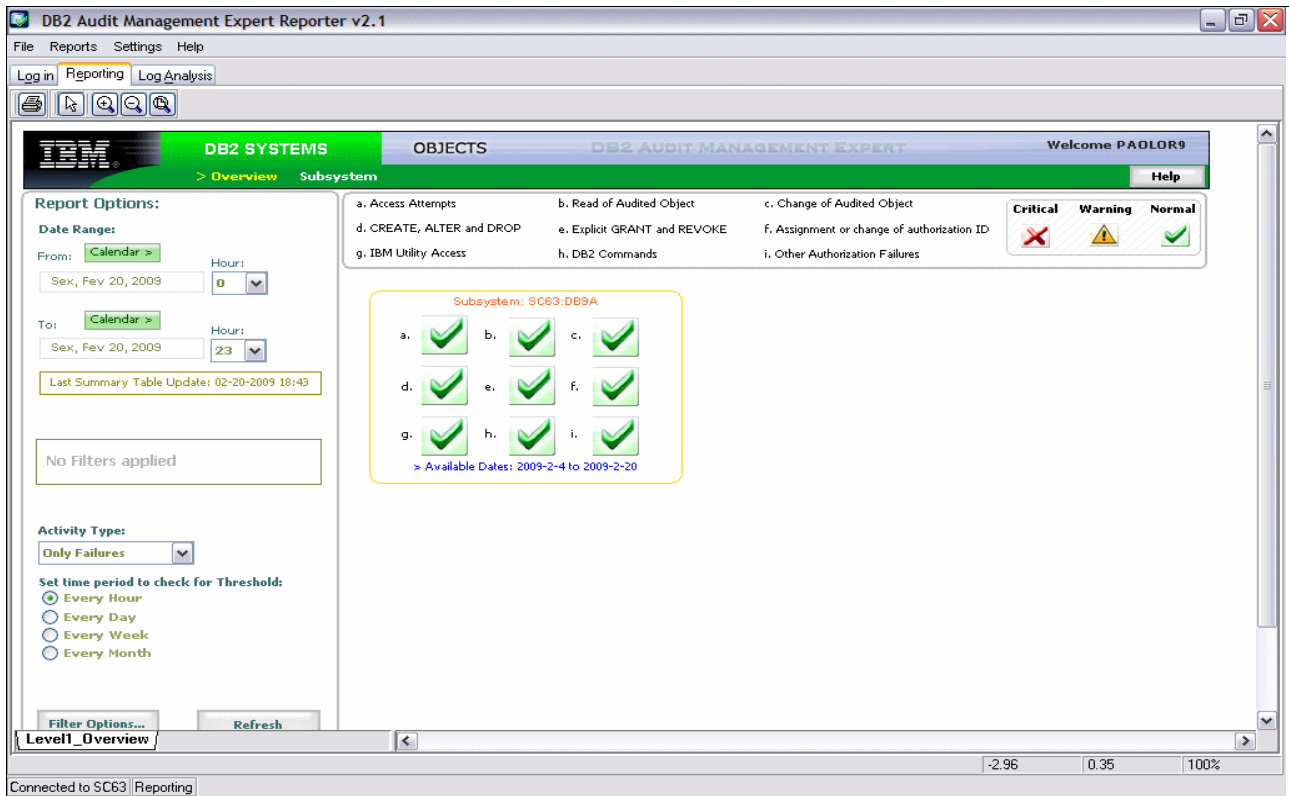


Figure 10-34 Selecting failures on DB2 subsystem (DB9A)

Click **a. Access Attempts** (Figure 10-35) to view unauthorized users failure trying to access a DB2 subsystem.

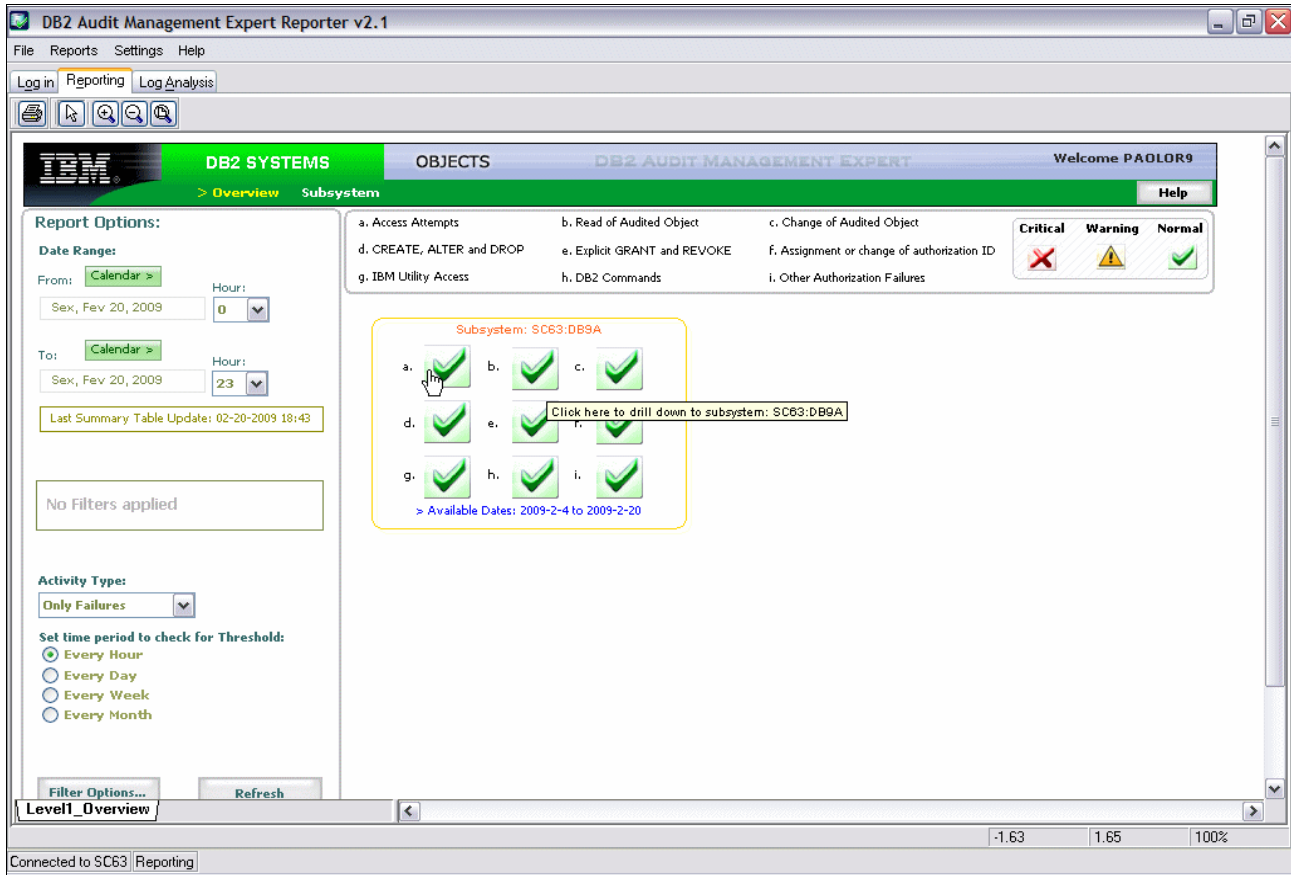


Figure 10-35 Access attempts failure

This window shows all failure activity by hour. Click **a. Access Attempts** (Figure 10-36).

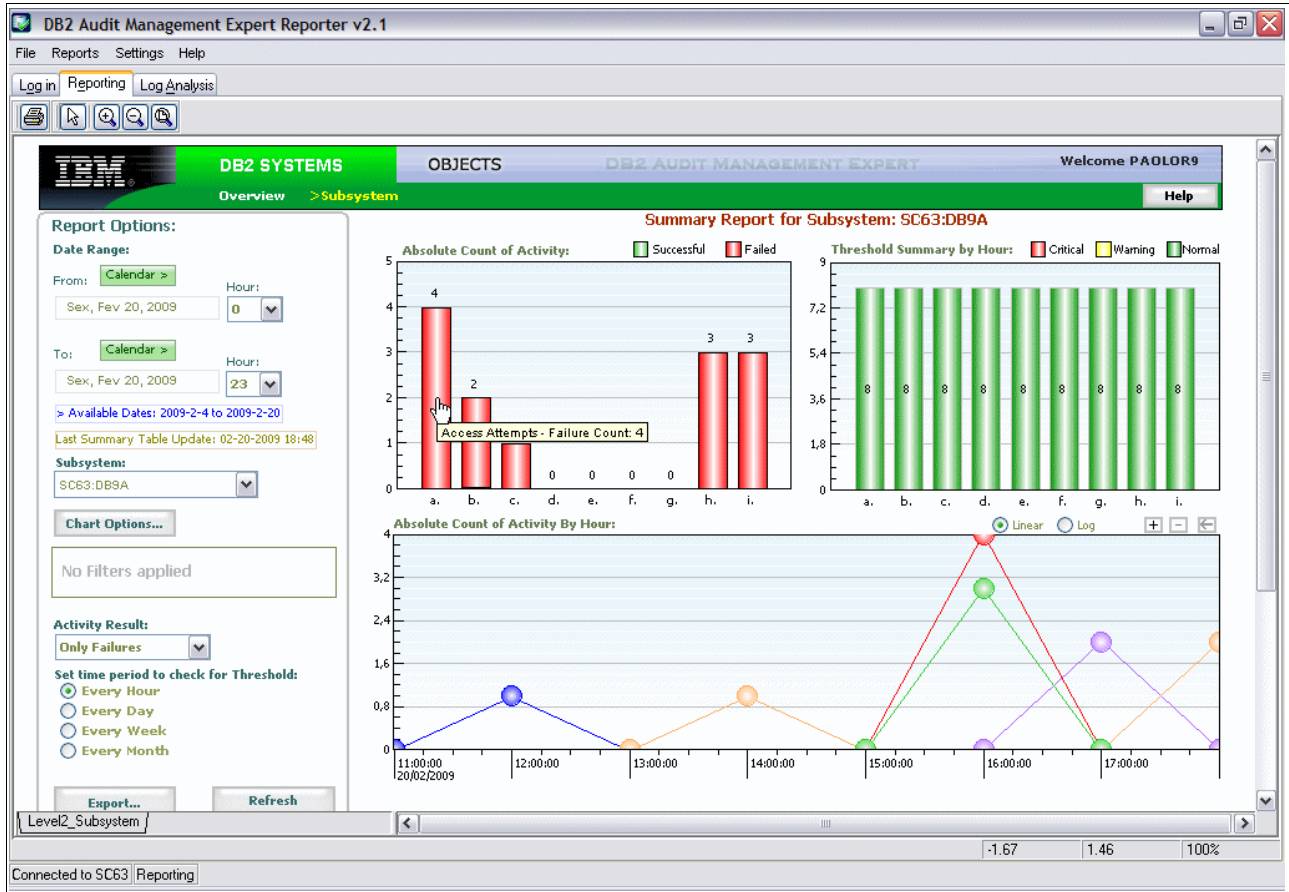


Figure 10-36 Summary Report of failure activity

Audit Management Expert shows four access attempts on DB2 subsystem (DB9A) system (SC63): 2 attempts by PAOLOR1 and 2 by PAOLOR2. See Figure 10-37.

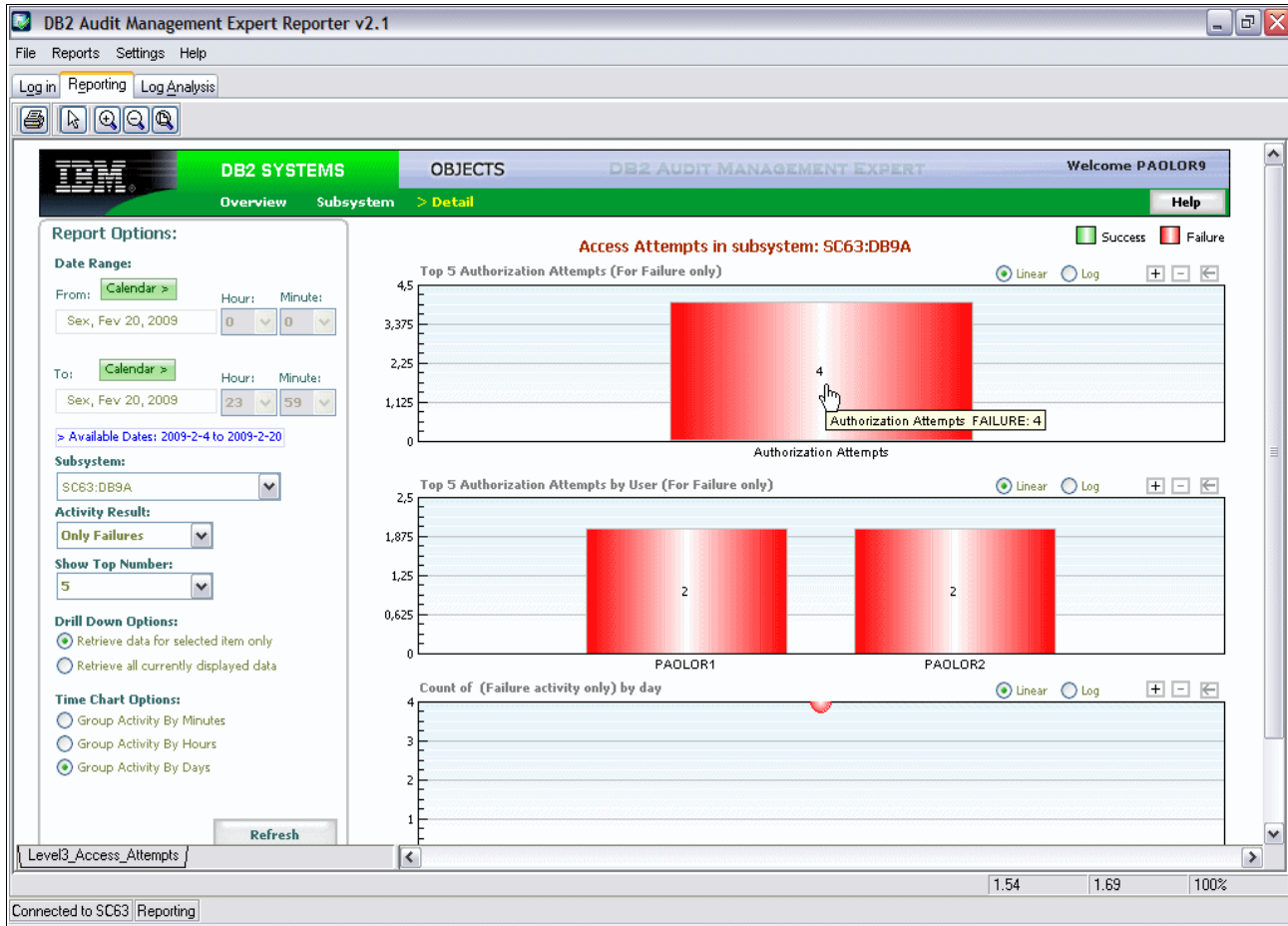


Figure 10-37 Access attempt

The auditors may ask when and what they try to execute. Click the chart to see detailed information about these accesses. See Figure 10-38.

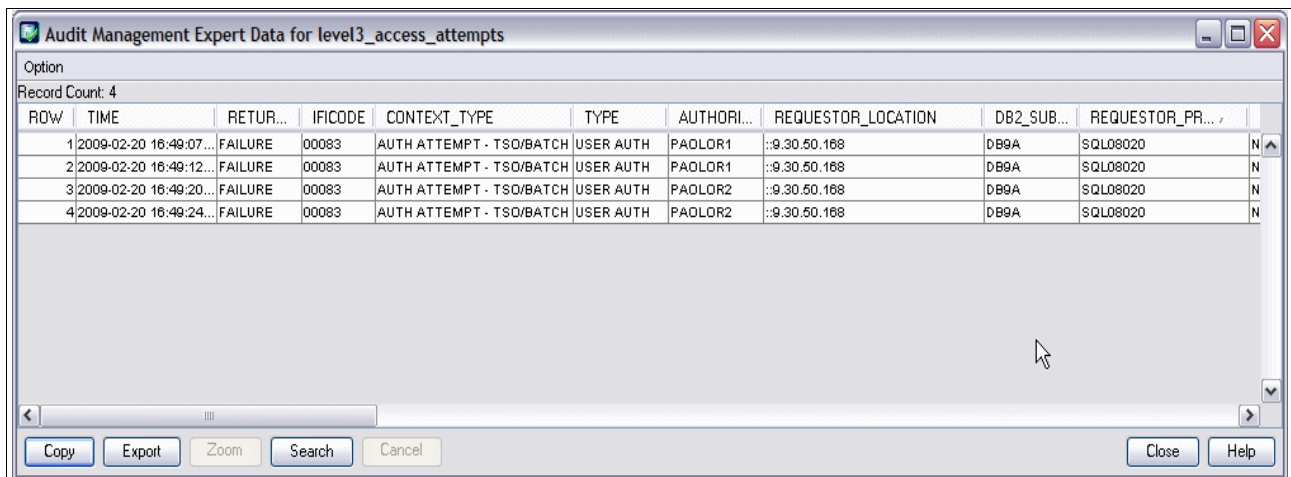


Figure 10-38 Access attempt - detail information

Report 2: Determine unauthorized users failure trying to set an unauthorized SQLID

We follow the same steps above:

- ▶ On DB2 SYSTEMS tab, select activity types **only failures** and provide the time interval to be reported. Click **Refresh**.
- ▶ Click in **a. Access Attempts**.

On the summary report of failure activity, choose **Other authorization failures**. See Figure 10-39.

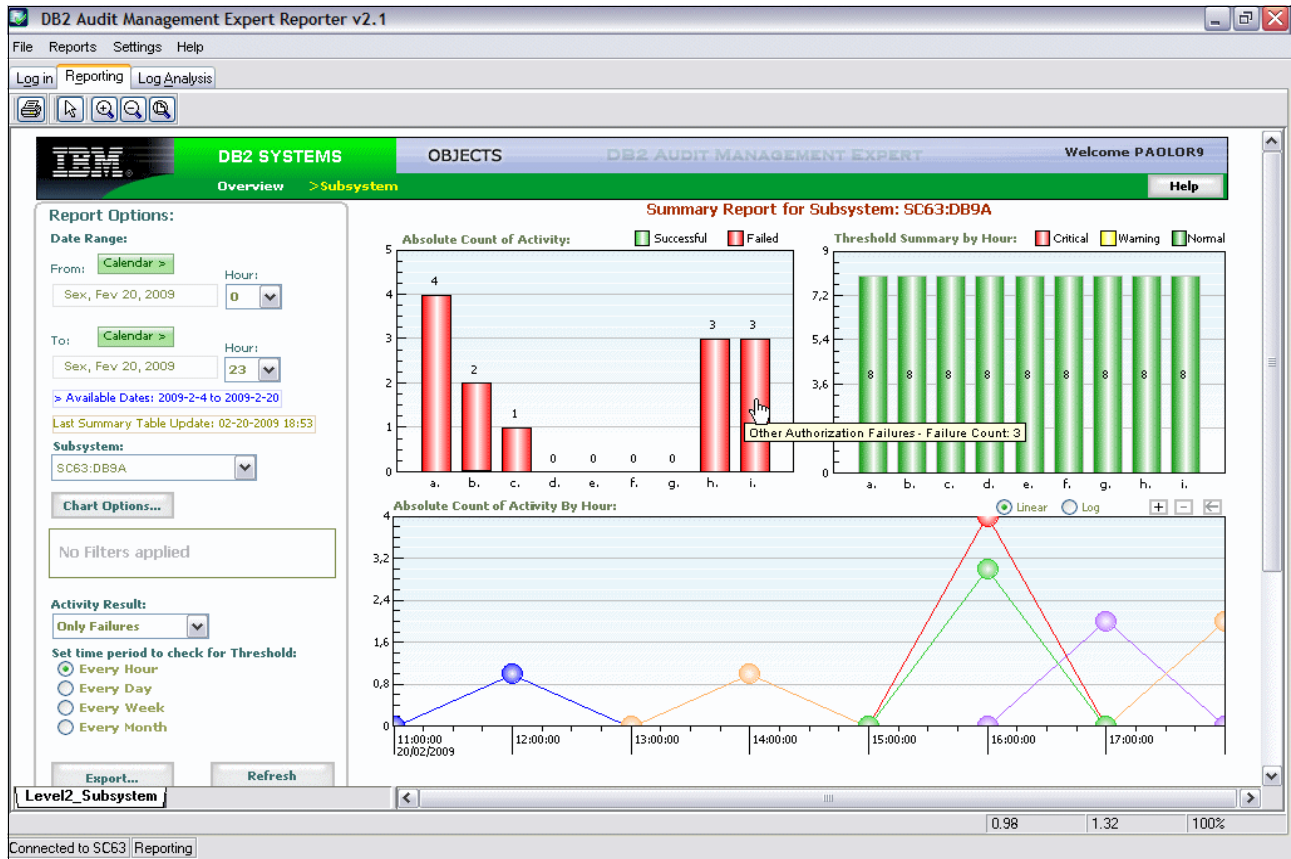


Figure 10-39 Summary Report of failure activity

This window shows that JOEUSER had three authorization failures.

What auditors want to know is what does JOEUSER execute? See Figure 10-40.

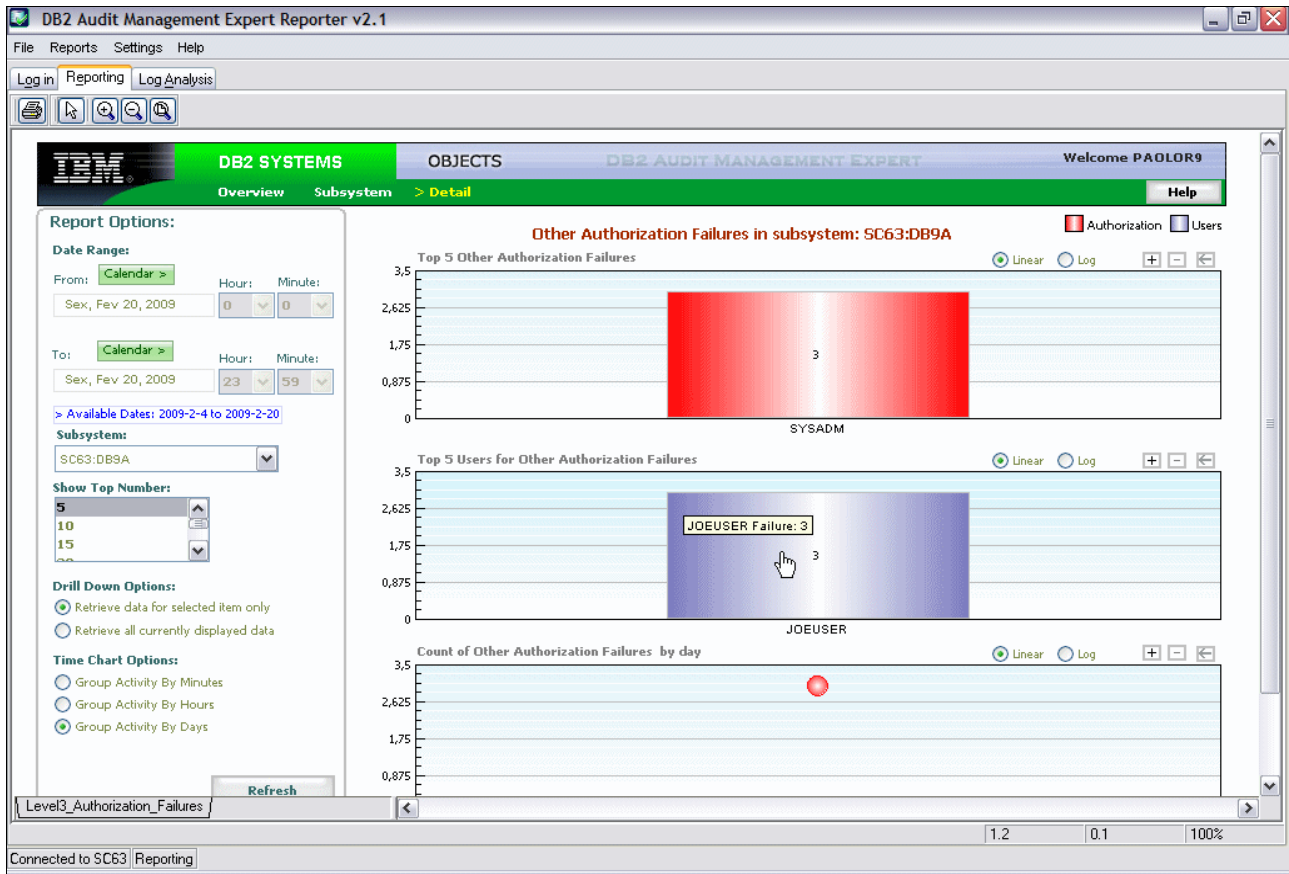


Figure 10-40 Other authorization failure window

JOEUSER tries to execute a SET CURRENT SQLID command and fails. See Figure 10-41.

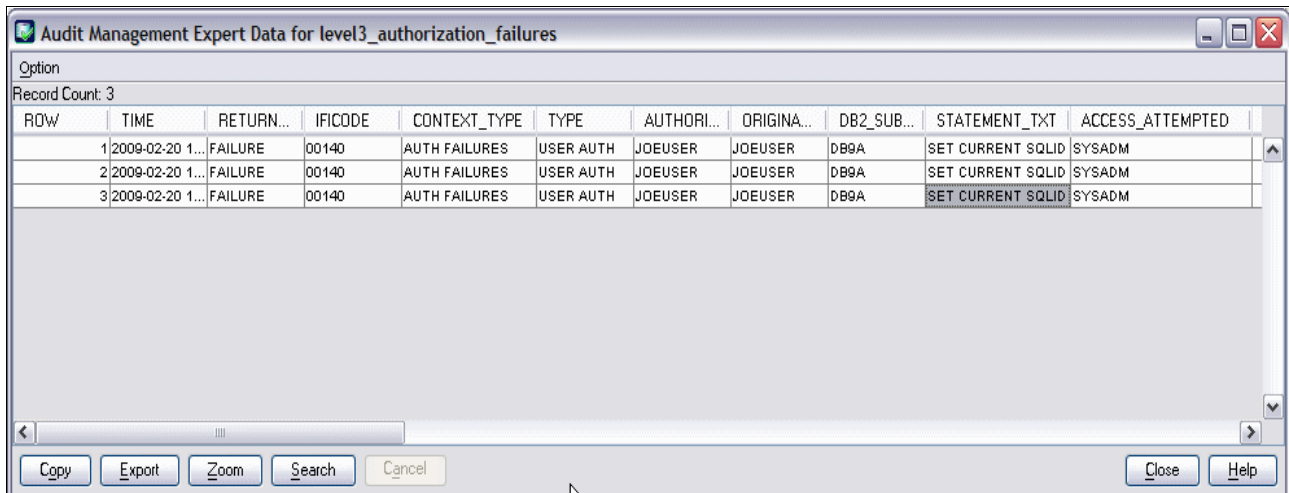


Figure 10-41 Command failure window

Report 3: Determine unauthorized users failure trying to select a monitored table

We follow the same steps above:

- ▶ On DB2 SYSTEMS tab, we select activity types **only failures** and provide the time interval to be reported. Click **Refresh**.
- ▶ Click in **a. Access Attempts**.

On the summary report of failure activity, choose **Read of audited objects**. See Figure 10-42.

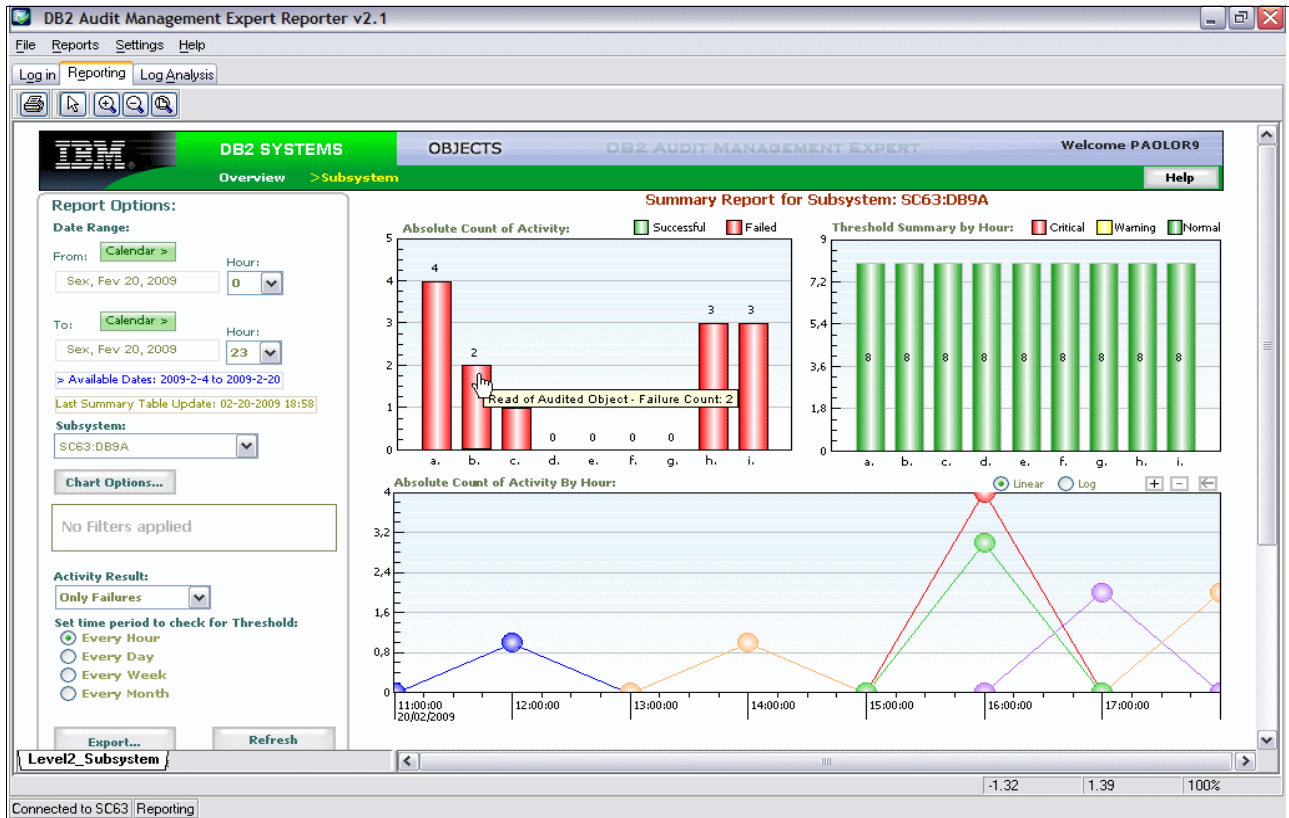


Figure 10-42 Summary Report of failure activity

This window shows that JOEUSER had two read of audited objects failures. See Figure 10-43 on page 251.

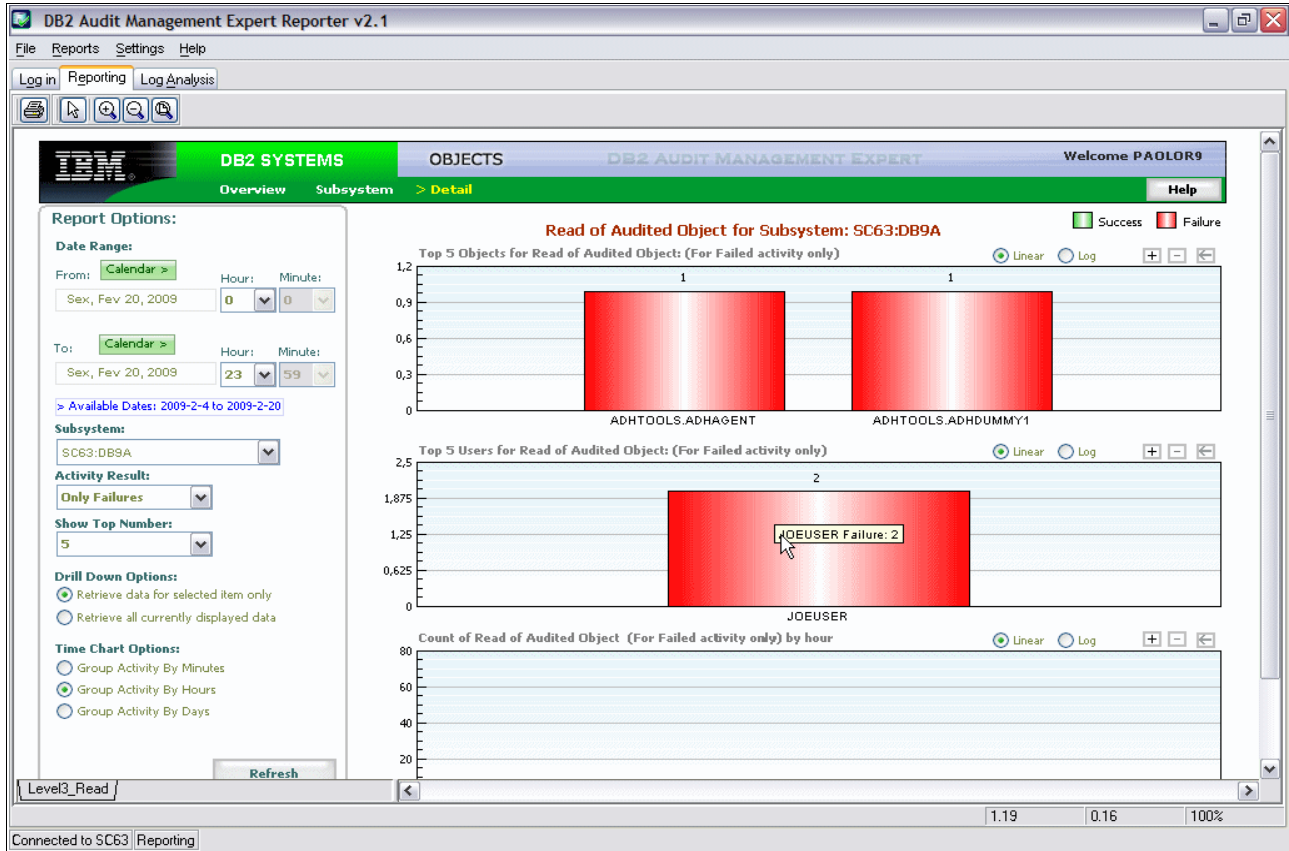


Figure 10-43 Read of audited object window

What auditors want to know is what did JOEUSER execute? See Figure 10-44.

ROW	TIME	DB2_SUB...	RETURN...	SCHEMA	NAME	TYPE	CONTEXT_T...	STATEMENT_TXT	AUTHORIZATIO...	ORIGINAL_OP_ID
1	2009-02-20 17:10:01.81...	DB9A	FAILURE	ADHTOOLS	ADHDUMMY1	TABLE/VIEW	AUTH FAILURES	SELECT * FROM "ADH...	JOEUSER	JOEUSER
2	2009-02-20 17:10:06.82...	DB9A	FAILURE	ADHTOOLS	ADHAGENT	TABLE/VIEW	AUTH FAILURES	SELECT * FROM "ADH...	JOEUSER	JOEUSER

Figure 10-44 SQL failure window

JOEUSER tried to execute a `SELECT * FROM ADH` and it failed. Auditors may ask for the complete SQL statement. See Figure 10-45.

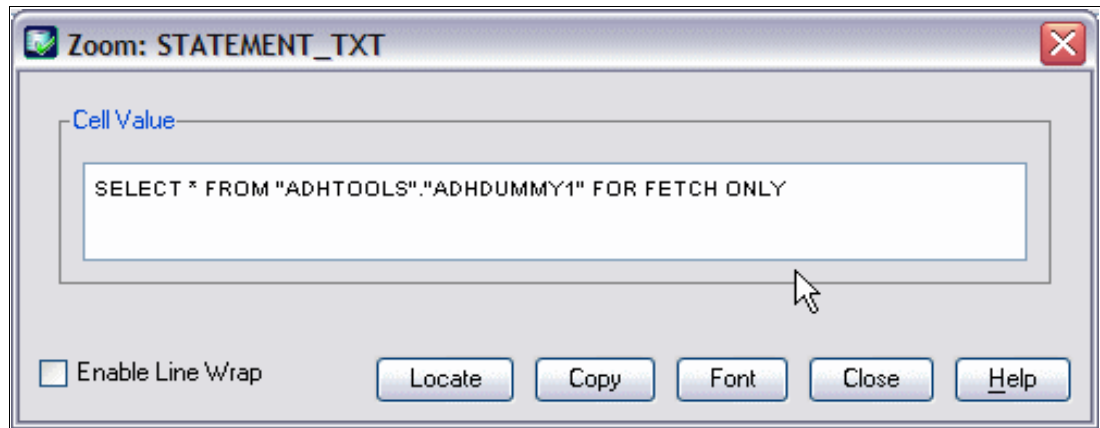


Figure 10-45 SQL statement

State actual results

The scenarios above guide auditors and allow them to run audit reports that are designed to present information useful for an audit without needing substantial knowledge of DB2 for z/OS or have any DB2 system privileges. Auditors are offered a historical perspective of data access. Audit Management Expert also gives the auditors independence so they can adhere to published industry standards without relying on personnel who are also being monitored.

The IBM Data Server Security Blueprint Audit Management Expert provides the mechanism to create, and access audit records. It also provides history records to determine what has happened in the past, which is often the only form of evidence to detect misuse.

10.2.4 Finding DDL activity

To eliminate encryption of a table, the privileged user has to follow these steps:

1. UNLOAD table.
2. DROP encrypted table.
3. CREATE unencrypted table.
4. LOAD table.

Audit Management Expert can produce a comprehensive report of DDL activity for auditors. It helps ensure that a encrypted table will stay encrypted.

In this scenario we describe how Audit Management Expert shows DD,L activity on a monitored DB2 subsystem. This report shows how to find create, alter and drop activity that has occurred on DB2 subsystems that are being monitored by Audit Management Expert.

Report

On the DB2 SYSTEMS tab we provide the time interval to be reported. Click Refresh, then click **d. CREATE, ALTER and DROP**. See Figure 10-46.

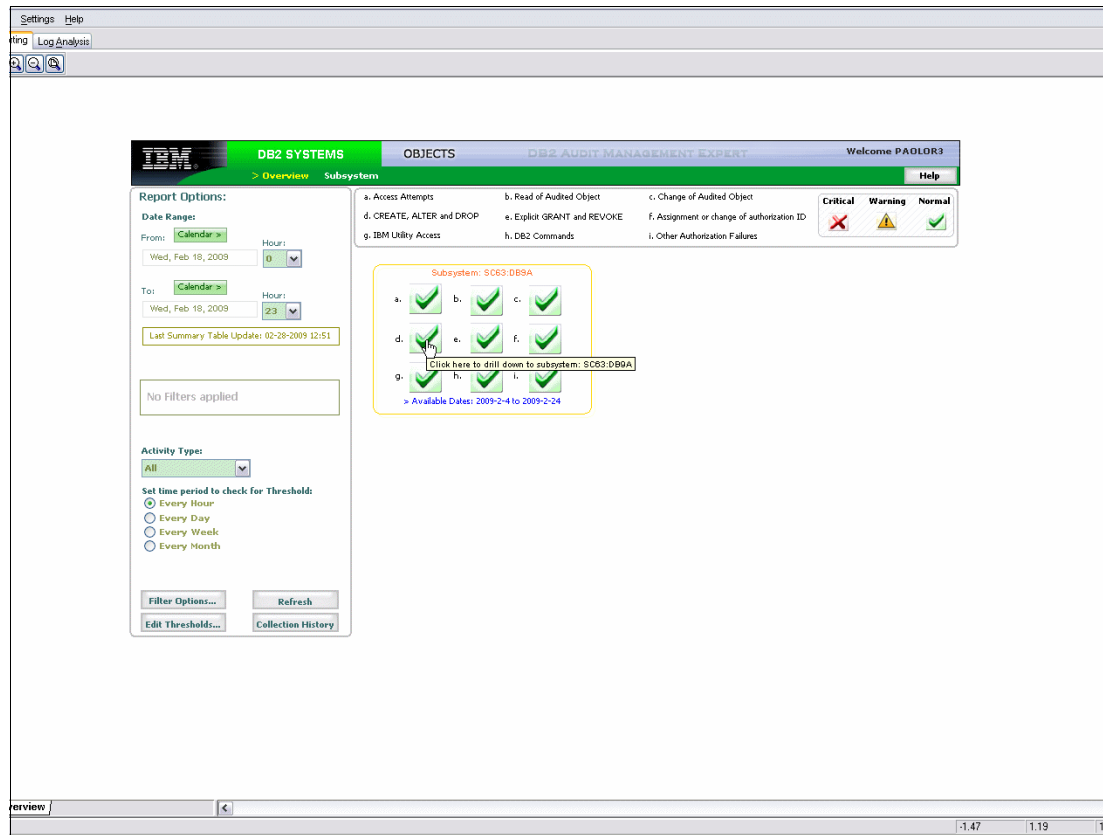


Figure 10-46 DB2 SYSTEM overview

Click **d. CREATE, ALTER, DROP** to view DDL activity. See Figure 10-47.

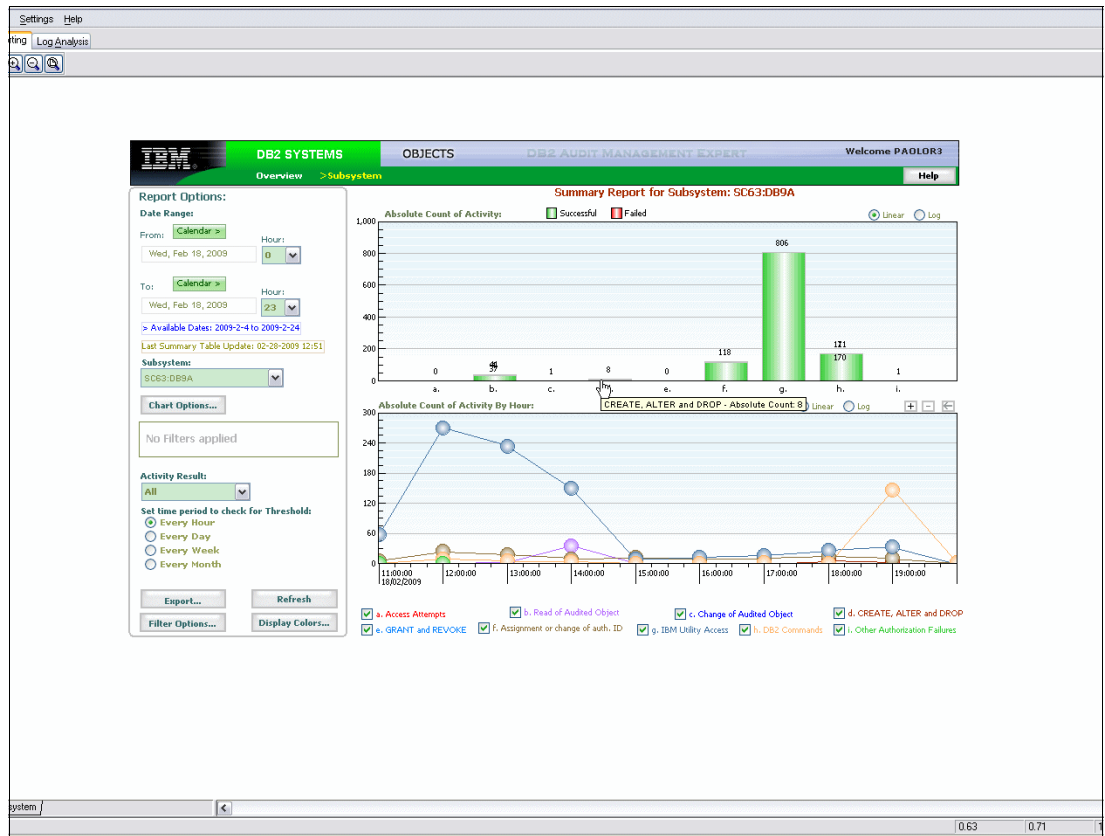


Figure 10-47 Activity overview

On this window we see that user PAOLOR5 executed four DROP and four CREATE. See Figure 10-48.

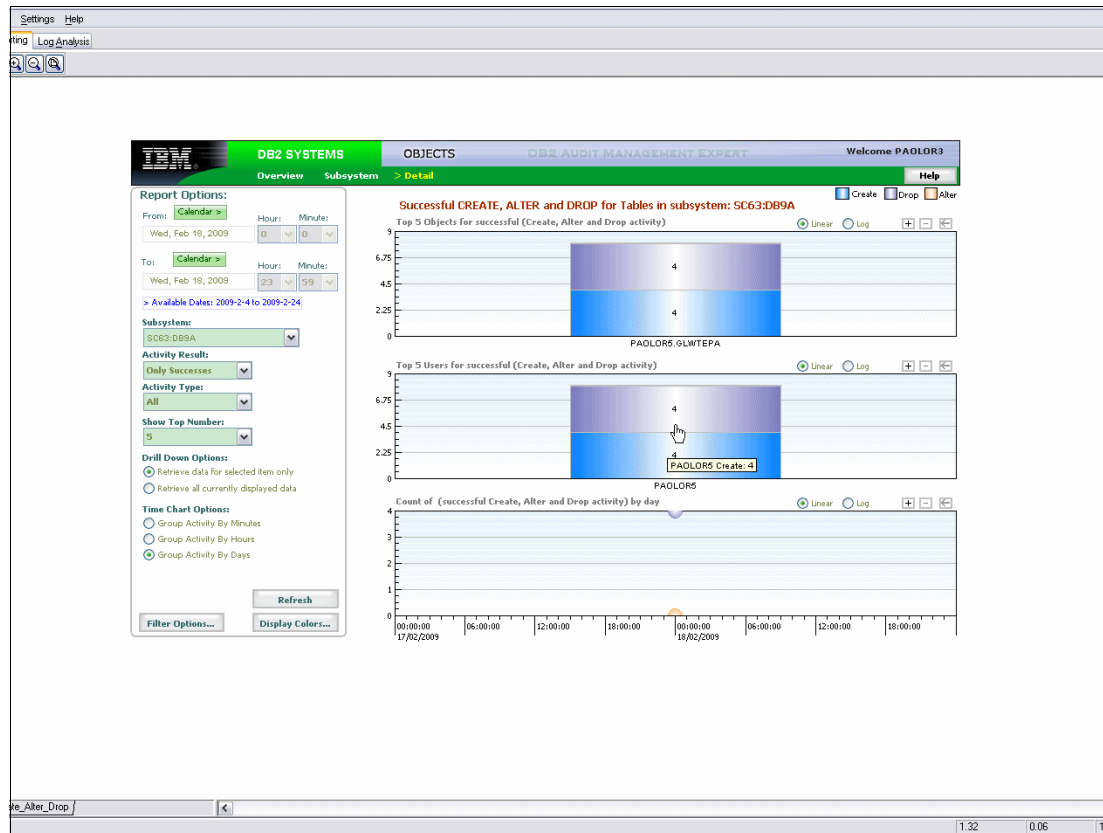


Figure 10-48 CREATE, ALTER, and DROP activity

Click the chart to get the list of executed DDL. See Figure 10-49.

The screenshot shows a window titled 'Audit Management Expert Data for level3_create_alter_drop'. It displays a table with 8 rows of data. The columns are: ROW, TIME, RETURN, SCHEMA, TYPE, CONTEX, NAME, STATEMENT_TXT, AUTHORIZATION, DB2_SUB, RESULT, NEW_SQ, CORREL, and IFCODE.

ROW	TIME	RETURN	SCHEMA	TYPE	CONTEX	NAME	STATEMENT_TXT	AUTHORIZATION	DB2_SUB	RESULT	NEW_SQ	CORREL	IFCODE
1	2009-02-18 1...	SUCCESS	PAOLOR5	TABLE/VIEW	DROP	QLWTEPA	DROP TABLESPACE "PAOLOR5".QLWSEPA"	PAOLOR5	DB9A	0/N/A	529124994/00142		
2	2009-02-18 1...	SUCCESS	PAOLOR5	TABLE/VIEW	CREATE	QLWTEPA	CREATE TABLE PAOLOR5.QLWTEPA (EMP_NO ...	PAOLOR5	DB9A	0/N/A	529124994/00142		
3	2009-02-18 1...	SUCCESS	PAOLOR5	TABLE/VIEW	DROP	QLWTEPA	DROP TABLESPACE "PAOLOR5".QLWSEPA"	PAOLOR5	DB9A	0/N/A	529121768/00142		
4	2009-02-18 1...	SUCCESS	PAOLOR5	TABLE/VIEW	CREATE	QLWTEPA	CREATE TABLE PAOLOR5.QLWTEPA (EMP_NO ...	PAOLOR5	DB9A	0/N/A	529121768/00142		
5	2009-02-18 1...	SUCCESS	PAOLOR5	TABLE/VIEW	DROP	QLWTEPA	DROP TABLESPACE "PAOLOR5".QLWSEPA"	PAOLOR5	DB9A	0/N/A	529121768/00142		
6	2009-02-18 1...	SUCCESS	PAOLOR5	TABLE/VIEW	CREATE	QLWTEPA	CREATE TABLE PAOLOR5.QLWTEPA (EMP_NO ...	PAOLOR5	DB9A	0/N/A	529121768/00142		
7	2009-02-18 1...	SUCCESS	PAOLOR5	TABLE/VIEW	DROP	QLWTEPA	DROP TABLESPACE "PAOLOR5".QLWSEPA"	PAOLOR5	DB9A	0/N/A	529123912/00142		
8	2009-02-18 1...	SUCCESS	PAOLOR5	TABLE/VIEW	CREATE	QLWTEPA	CREATE TABLE PAOLOR5.QLWTEPA (EMP_NO ...	PAOLOR5	DB9A	0/N/A	529123912/00142		

Figure 10-49 List of executed DDL

Click the statement to get the DROP statement executed. See Figure 10-50.

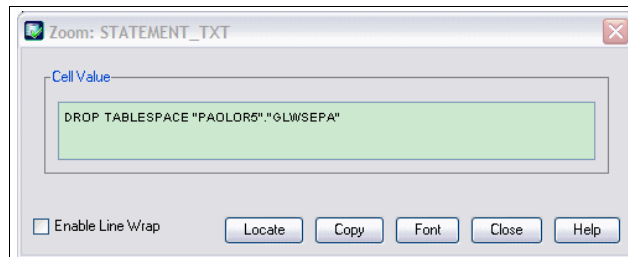


Figure 10-50 DROP detail

We go back to the list of executed DDL shown in Figure 10-49 on page 255 and shown in Figure 10-51 (both show the list of executed DDL).

The window "Audit Management Expert Data for level3_create_alter_drop" displays a table with 8 records. The columns include ROW#, TIME, RETURN..., SCHEMA, TYPE, CONTEX..., NAME, STATEMENT_TXT, AUTHORIZATION..., DB2_SUB..., RESULT, NEW_SEQ..., CORREL..., and IFICODE.

ROW#	TIME	RETURN...	SCHEMA	TYPE	CONTEX...	NAME	STATEMENT_TXT	AUTHORIZATION...	DB2_SUB...	RESULT	NEW_SEQ...	CORREL...	IFICODE
1	2009-02-18 1...	SUCCESS	PAOLORS	TABLEVIEW	DROP	GLWTEPA	DROP TABLESPACE "PAOLORS"."GLWSEPA"	PAOLORS	DB9A	0	N/A	529124984	00142
2	2009-02-18 1...	SUCCESS	PAOLORS	TABLEVIEW	CREATE	GLWTEPA	CREATE TABLE PAOLORS.GLWTEPA (EMP_NO ...	PAOLORS	DB9A	0	N/A	529124984	00142
3	2009-02-18 1...	SUCCESS	PAOLORS	TABLEVIEW	DROP	GLWTEPA	DROP TABLESPACE "PAOLORS"."GLWSEPA"	PAOLORS	DB9A	0	N/A	529121768	00142
4	2009-02-18 1...	SUCCESS	PAOLORS	TABLEVIEW	CREATE	GLWTEPA	CREATE TABLE PAOLORS.GLWTEPA (EMP_NO ...	PAOLORS	DB9A	0	N/A	529121768	00142
5	2009-02-18 1...	SUCCESS	PAOLORS	TABLEVIEW	DROP	GLWTEPA	DROP TABLESPACE "PAOLORS"."GLWSEPA"	PAOLORS	DB9A	0	N/A	529121768	00142
6	2009-02-18 1...	SUCCESS	PAOLORS	TABLEVIEW	CREATE	GLWTEPA	CREATE TABLE PAOLORS.GLWTEPA (EMP_NO ...	PAOLORS	DB9A	0	N/A	529121768	00142
7	2009-02-18 1...	SUCCESS	PAOLORS	TABLEVIEW	DROP	GLWTEPA	DROP TABLESPACE "PAOLORS"."GLWSEPA"	PAOLORS	DB9A	0	N/A	529123912	00142
8	2009-02-18 1...	SUCCESS	PAOLORS	TABLEVIEW	CREATE	GLWTEPA	CREATE TABLE PAOLORS.GLWTEPA (EMP_NO ...	PAOLORS	DB9A	0	N/A	529123912	00142

Figure 10-51 List of executed DDL

Click the CREATE statement to execute the create statement. See Figure 10-52.

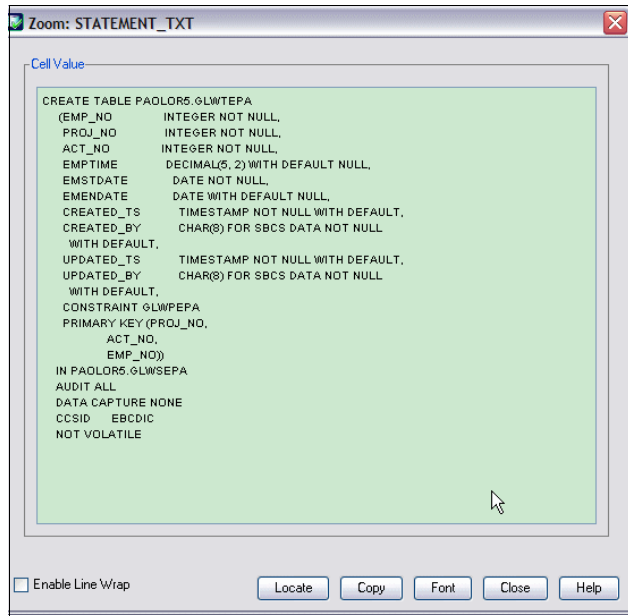


Figure 10-52 CREATE detail

State actual results

The scenarios above guide auditors in Audit Management Expert window and allows them to run audit reports, which are designed to present information in a useful way for an audit without developing substantial knowledge of DB2 for z/OS or have any DB2 system privileges. Auditors are able to access a historical perspective of DB2 structure changes. Audit Management Expert also gives the auditors independence so they can verify structure changes without relying on personnel who are also being monitored.

10.3 Log Analysis User Interface

Log Analysis in Audit Management Expert for z/OS allows easy access to information stored on DB2 logs. It enables you to view who modified audited tables in a DB2 system and to see the changes made to the tables.

DB2 logs

When a program modifies DB2 data, this happens in main storage, in the buffer pool. Writing those changes to disk can happen asynchronously with the unit of work at any time, but generally after the changes have been subject to COMMIT or ROLLBACK. The I/O to the Log data set at COMMIT/ROLLBACK time is instead a synchronous process.

When an asynchronous write is done to a table space on disk, the corresponding log information is first written to the log data set. The log data set is the master copy of the data. It allows DB2 to recover from whatever problem might happen in the system

As you make changes to your tables, DB2 writes appropriate records to the DB2 log allowing DB2 to back out of the changes if a unit of recovery fails, or to apply these changes during a recovery.

The DB2 log is mapped onto data sets. Each DB2 subsystem has a predefined fixed set of active log data sets on disk. Log records are first written by DB2 into a log buffer. They are subsequently written on to the active log data sets. As an active log data set fills up, DB2 moves onto the next active log data set.

When all active log data sets have been filled, DB2 wraps around to the first active log data set and uses it again. In order not to lose log records that may be required for a backout or recovery, the active log data sets are automatically offloaded as they fill up. They are offloaded by DB2 to archive log data sets that may be on disk or on tapes. In contrast to the set of 99 active log data sets, there is a set of 10000 archive log data sets. The individual log data sets, both active and archive, are recorded in the DB2-managed Bootstrap Data Set (BSDS). Only those archive logs that are still needed for backout or recovery operations need to be kept.

10.3.1 Generating Log Analysis reports

This section provides an overview of the Audit Management Expert for z/OS log analysis feature. Audit Management Expert Log analysis provides answers to the following questions:

- ▶ Who updated the audited table?
- ▶ What data was updated?

From the DB2 Audit Management Expert Reporter main window, click the **Log Analysis** tab. The “Welcome” page of the Log Analysis Advisor displays (Figure 10-53).

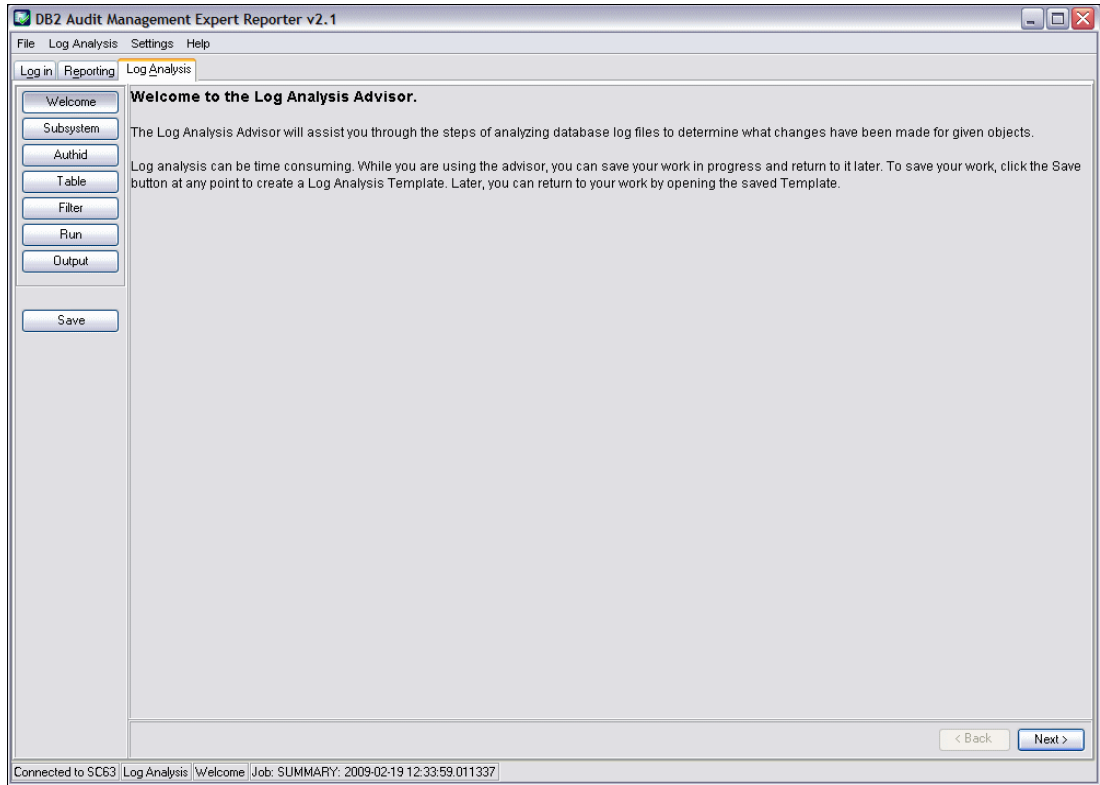


Figure 10-53 Log Analysis advisor “Welcome” window

Note: The Log Analysis tab is disabled in the reporting interface if you do not have permission to run Log Analysis jobs.

Click **Next**. The Subsystem page displays (Figure 10-54).

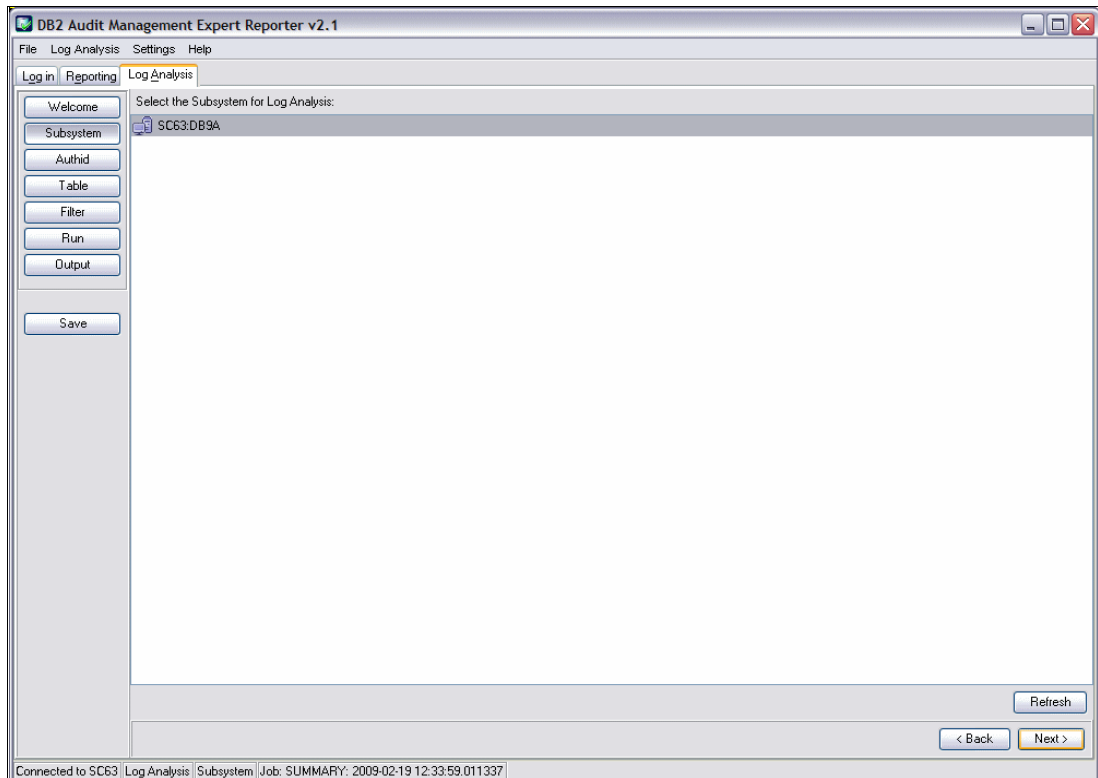


Figure 10-54 Select Subsystem for Log Analysis

Select a subsystem and click **Next**. The AuthID page appears. In the list we have included user PAOLOR3, because we want to see what PAOLOR3 executed (Figure 10-55).

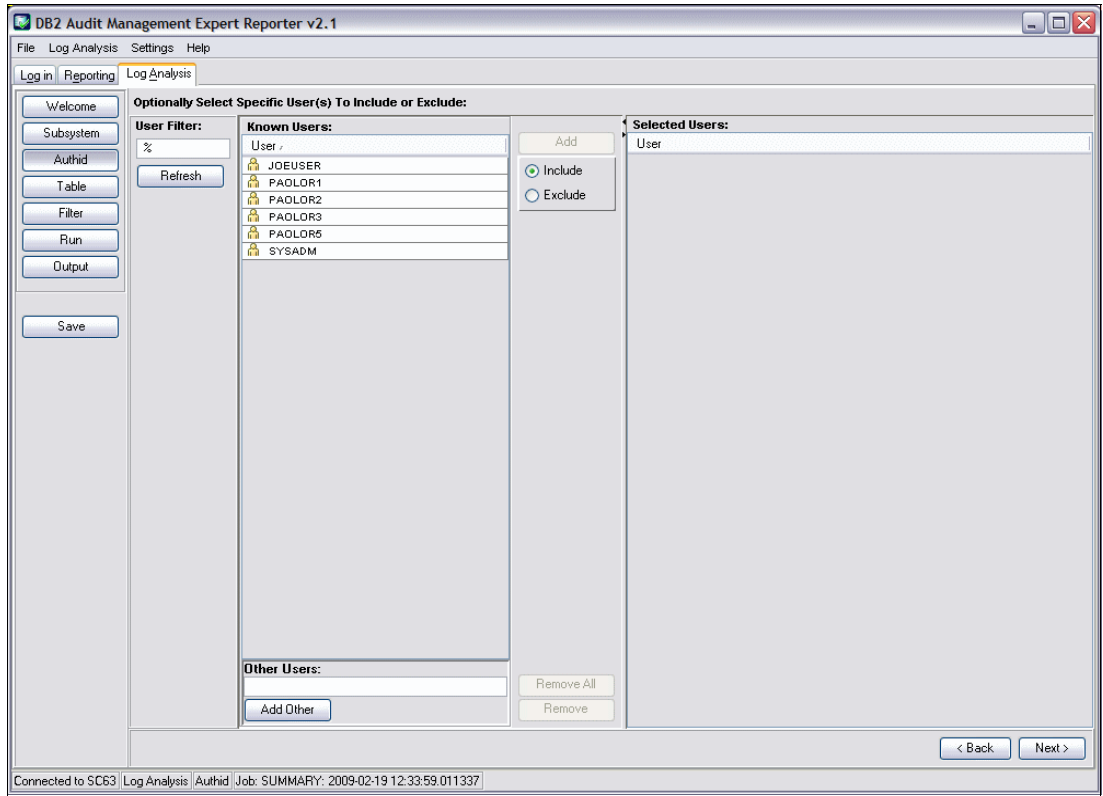


Figure 10-55 Generate User List

Click **Next**. The table page displays. Select a table. To filter the list of available tables, specify schema criteria in the schema field and table name criteria in the name field and click **Refresh**. In this page we want to see the GLWWRK1 schema. See Figure 10-56.

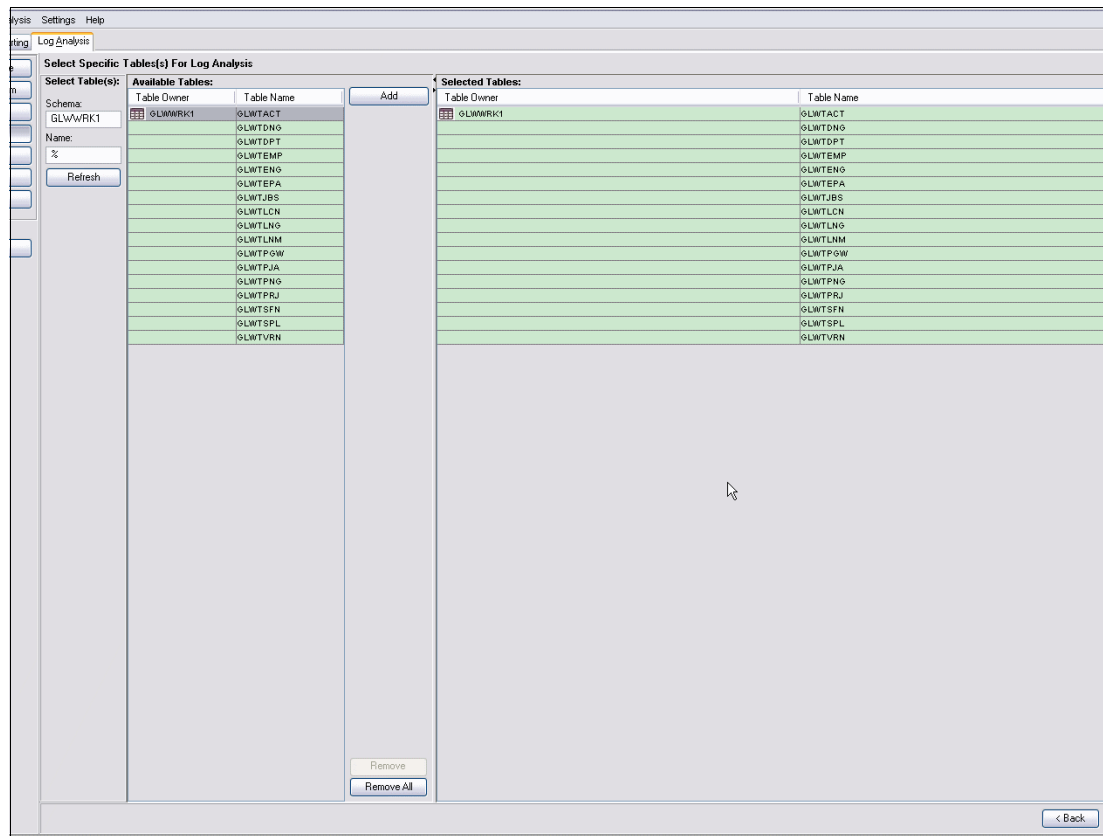


Figure 10-56 Add table list

Click **Next**. The filter page appears. By default, the Log Range date and time values are set from the time you log into the reporting client. Specify a log range and specify at least one statement type. In our case, we want to audit inserts, updates, and deletes. See Figure 10-57.

Summary reports are always generated. You can generate a details report in addition to the Summary report by choosing the **Detailed Activity Report** option.

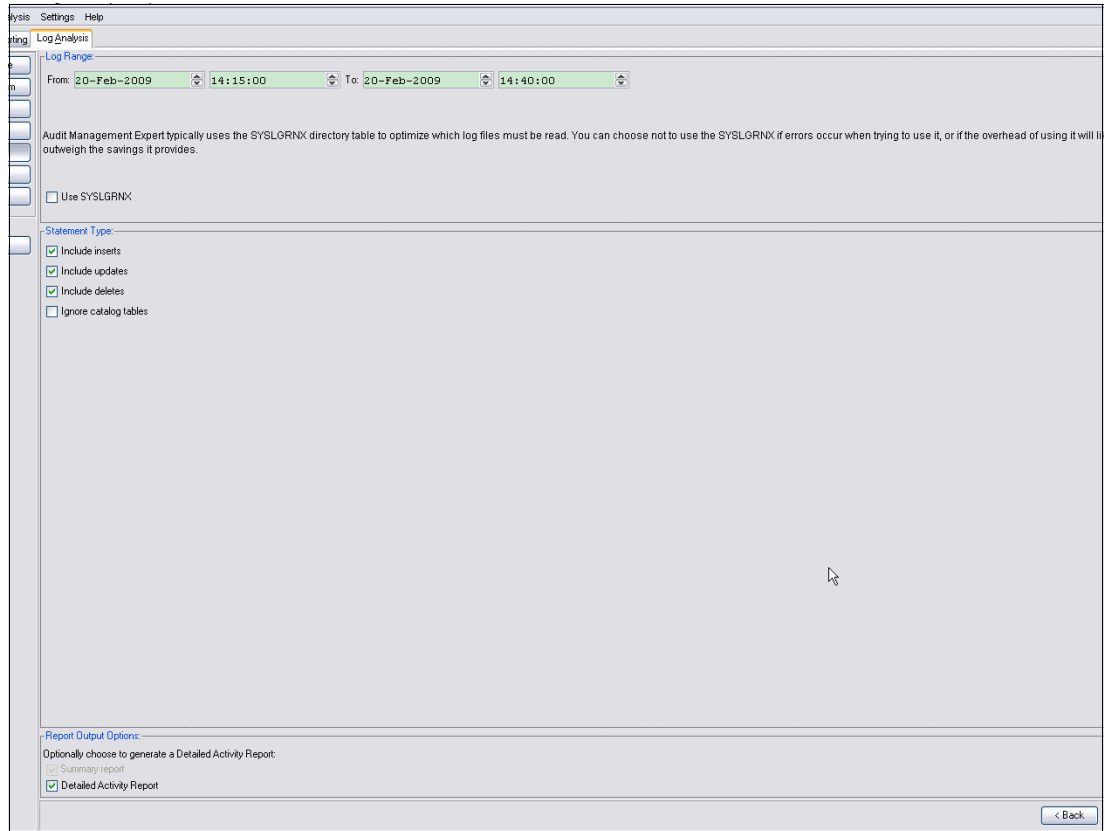


Figure 10-57 Add statement

Click **Next**. On this page, click **Generate JCL** to generate log analysis JCL. Click **Run** to run the job. See Figure 10-58.

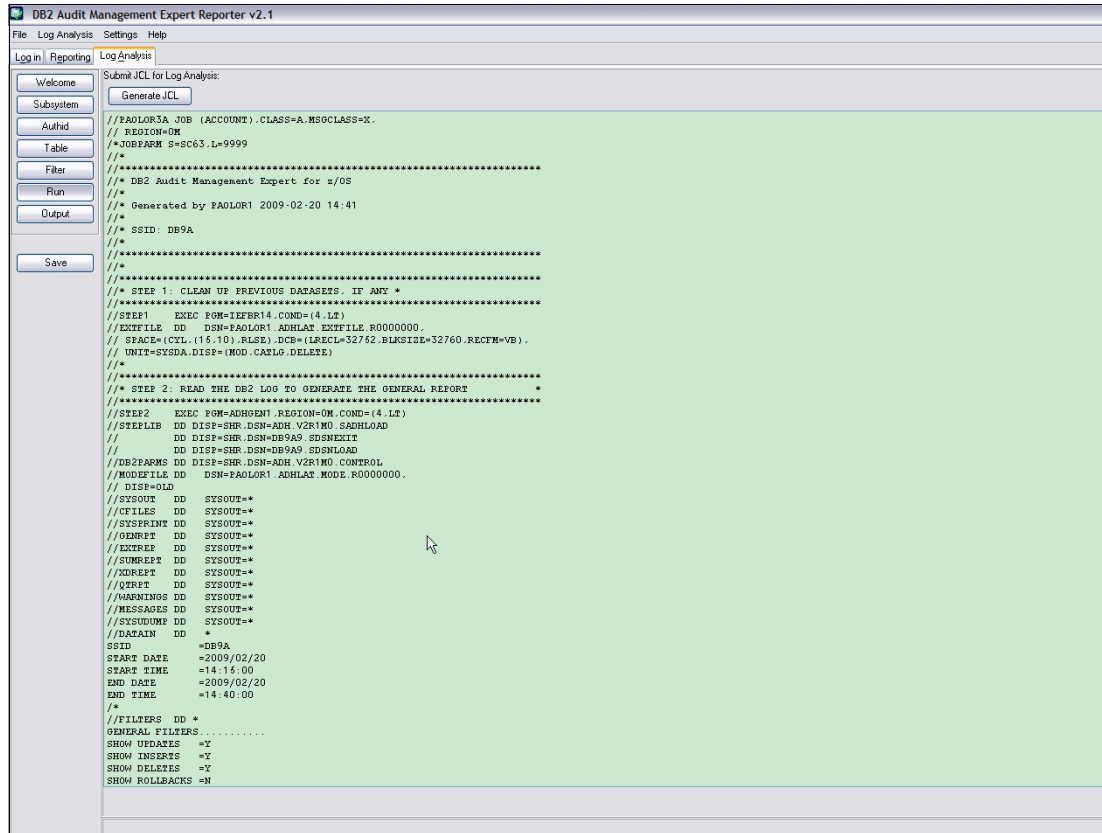


Figure 10-58 Generate JCL

Summary report

The summary report presents summary information about the table space, database, and number of updates, inserts, and deletes for a table. Most data in the report is self-explanatory and comes directly from the DB2 log record. See Figure 10-59.

Submitted Log Analysis Jobs:

Name	Report Type	Status	Job ID	MAX CC	Start Time	Last Updated	User	Subsystem
PAULUH3A	Detail	Completed	JOB24561	0	20 February 2009 13:17	20 February 2009 13:17	PAULUH3	SC63 DB9A
PAOLDR3A	Detail	Completed	JOB24532	0	20 February 2009 12:37	20 February 2009 12:37	PAOLDR3	SC63 DB9A
PAOLDR3A	Detail	Failed	JOB24768	8	20 February 2009 16:24	20 February 2009 16:25	PAOLDR3	SC63 DB9A
PAOLDR3A	Detail	Completed	JOB24617	0	20 February 2009 14:41	20 February 2009 14:42	PAOLDR3	SC63 DB9A
PAOLDR3A	Summary	Completed	JOB24764	0	20 February 2009 16:22	20 February 2009 16:23	PAOLDR3	SC63 DB9A

Report Output:

```

DB2 LOG ANALYSIS - SUMMARY REPORT: DB9A
*****
LOG RANGE
*****
START DATE   : 2009/02/20
START TIME   : 14:15:00
END DATE     : 2009/02/20
END TIME     : 14:40:00

FILTERS
*****
SHOW UPDATES : Y
SHOW INSERTS : Y
SHOW DELETES : Y
SHOW ROWLOCKS : N
CATALOG DATA : N

INCLUDE-TABLE ..... GLAWRKT1 GLAWTACT
INCLUDE-TABLE ..... GLAWRKT1 GLAWINDG
INCLUDE-TABLE ..... GLAWRKT1 GLAWIDFG
INCLUDE-TABLE ..... GLAWRKT1 GLAWIDFE
INCLUDE-TABLE ..... GLAWRKT1 GLAWIDKE
INCLUDE-TABLE ..... GLAWRKT1 GLAWIDNG
INCLUDE-TABLE ..... GLAWRKT1 GLAWIDEA
INCLUDE-TABLE ..... GLAWRKT1 GLAWIDCS
INCLUDE-TABLE ..... GLAWRKT1 GLAWIDCN
INCLUDE-TABLE ..... GLAWRKT1 GLAWIDLN
INCLUDE-TABLE ..... GLAWRKT1 GLAWIDGM
INCLUDE-TABLE ..... GLAWRKT1 GLAWIDJA
INCLUDE-TABLE ..... GLAWRKT1 GLAWIDJG
INCLUDE-TABLE ..... GLAWRKT1 GLAWIDJF
INCLUDE-TABLE ..... GLAWRKT1 GLAWIDJN
INCLUDE-TABLE ..... GLAWRKT1 GLAWIDJN
INCLUDE-TABLE ..... GLAWRKT1 GLAWIDJN

*****
* COMMITTED ACTIVITY
*****
OBJECT TYPE/NAME ..... UPDATES  INSERTS  DELETES  MD
-----
TABLE ..... GLAWRKT1 GLAWIDVFN  1         1         0
TABLE ..... GLAWRKT1 GLAWIDGM    0        105        0
TABLE ..... GLAWRKT1 GLAWIDCN    0        130        0
TABLE ..... GLAWRKT1 GLAWIDCS    0         15         0
  
```

Figure 10-59 Summary report

Details report

Each details report, from ACTION to the last column reported on, represents a row modification to a table. The report shows three row images at different points in time. Most data in the report comes directly from the DB2 log record. See Figure 10-60.

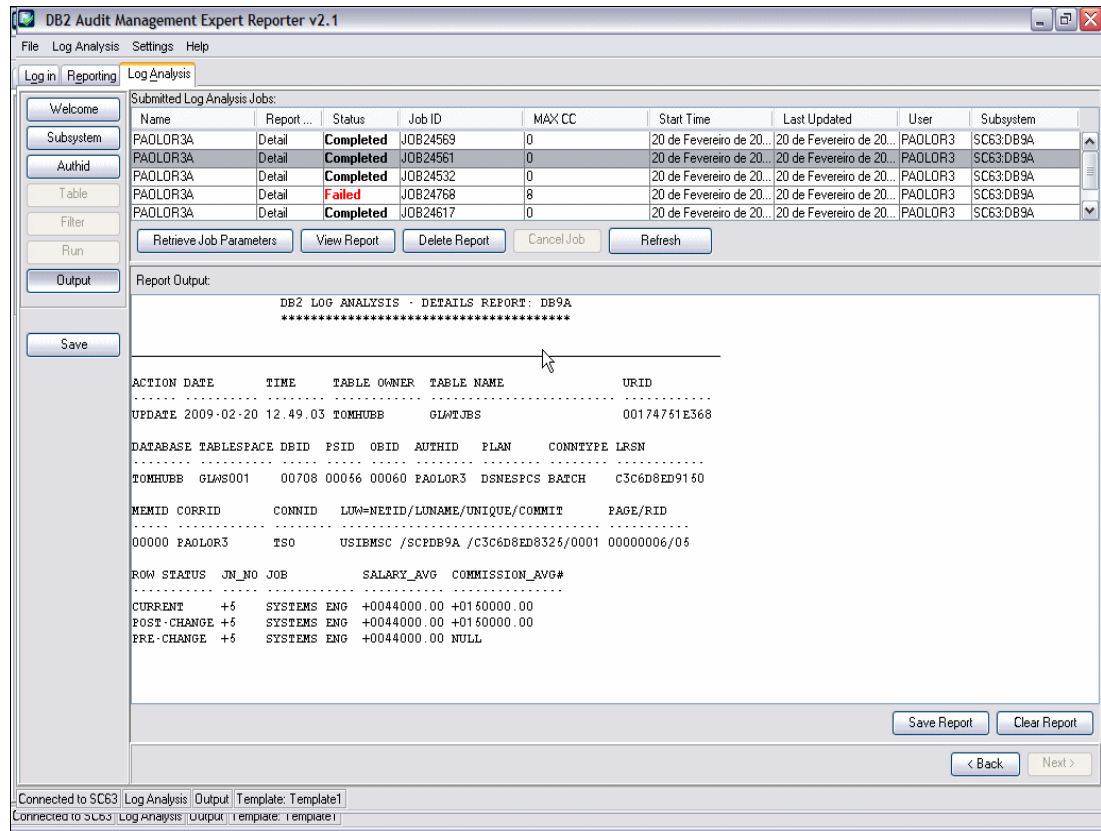


Figure 10-60 Detail report

The LRSN column in the report contains a hexadecimal display value of the log record time stamp. This field represents the actual time or sequence number of the log record creation, and is used mostly in data sharing environments to synchronize log records across members of a data sharing group.

Viewing Log Analysis reports

From the Log Analysis Advisor Output window, you can view log analysis reports. See Figure 10-61.

To view reports, from the Output page, click **Refresh** to update the Submitted Log Analysis Jobs list. Click the job in the Submitted Log Analysis Jobs list that has the report output you want to view. Click **View Report**.

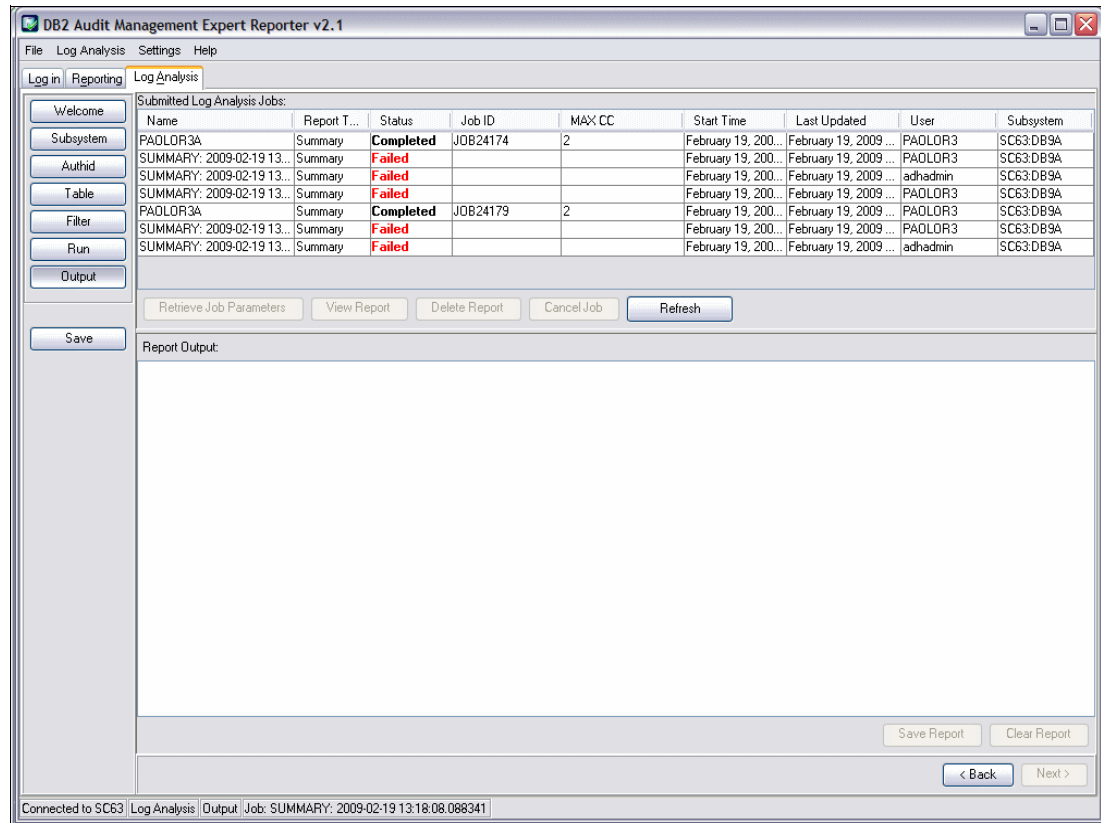


Figure 10-61 View log

Saving Log Analysis reports

From the “Log Analysis Advisor Output” window, we can save Log Analysis reports locally.

To save a Log Analysis report, from the Output page, click **Save Report**. The “Export Report to txt File” window appears. Navigate to the folder in which you want to save the report and in the File name field, type a name for the report. Click **Save**.

Deleting a Log Analysis report

From the “Log Analysis Advisor Output” window, we can delete Log Analysis reports.

Deleting a report removes the job information.

To delete a report, from the Output page, click **Refresh** to update the Submitted Log Analysis Jobs list, then click the job in the Submitted Log Analysis Jobs list that contains the report you want to delete. Click **Delete Report**.

Canceling a Log Analysis job

From the “Log Analysis Advisor Output” window, we can cancel a Log Analysis job.

To cancel a job, from the Output page, click **Refresh** to update the Submitted Log Analysis Jobs list then click the job in the Submitted Log Analysis Jobs list that you want to cancel. Click **Cancel job**.

Retrieving Log Analysis job parameters

From the “Log Analysis Advisor Output” window, we can retrieve Log Analysis job parameters.

To retrieve job parameters, from the Output page, click **Refresh** to update the Submitted Log Analysis Jobs list then click the job in the Submitted Log Analysis Jobs list that contains the job parameters you want to retrieve. Click **Retrieve Job Parameters**.

10.3.2 Templates and jobs

Within the Log Analysis Advisor, we work with templates and jobs:

- ▶ **Templates**

These are the parameters that are set in the advisor, including the JCL, that can be saved at any time by clicking **Save**, or by using the Log Analysis Save Template menu option.

- ▶ **Jobs**

After you submit the JCL, the parameters are saved as a job and are listed on the Output page.

Opening a Log Analysis template

To open a Log Analysis template, from the Log Analysis menu, click **Open template**. The Open Template window displays. In the Saved Templates list, click the template you want to open. Click **Retrieve Template** to open the selected template. See Figure 10-62.

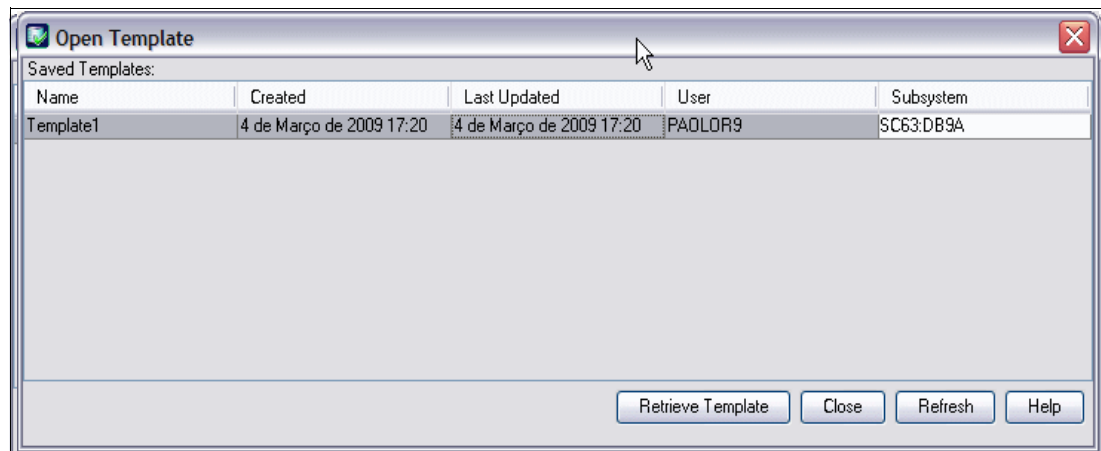


Figure 10-62 Open Template

Saving a Log Analysis template

This information describes how to save a template using the **Log Analysis → Save Template or Log Analysis → Save Template As** menu option. You can save a template at any point in the Log Analysis process. At a later time, you can return to your work by opening the saved template.

To save a Log Analysis template, from the Log Analysis menu, click **Save Template** or **Save Template As**. The Save Template window displays. In the Name field, type a name for the template. Click **Save** to save the template to the Audit Management Expert server. See Figure 10-63.

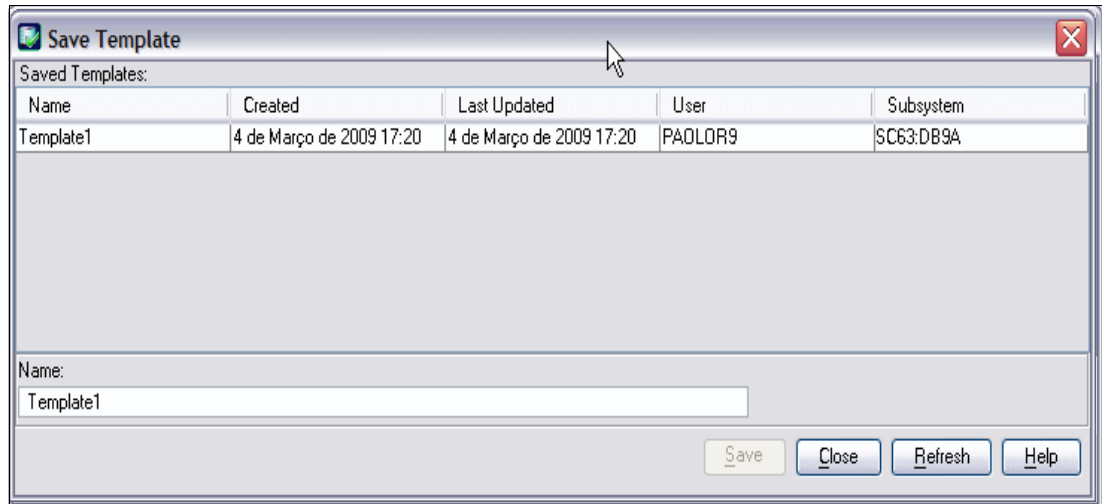


Figure 10-63 Save Template

Deleting a Log Analysis template

To delete a Log Analysis template, from the Log Analysis menu, click **Delete Template**. The Delete Template window displays. In the Saved Templates list, click the template you want to delete. Click **Delete**. The template is deleted. See Figure 10-64.

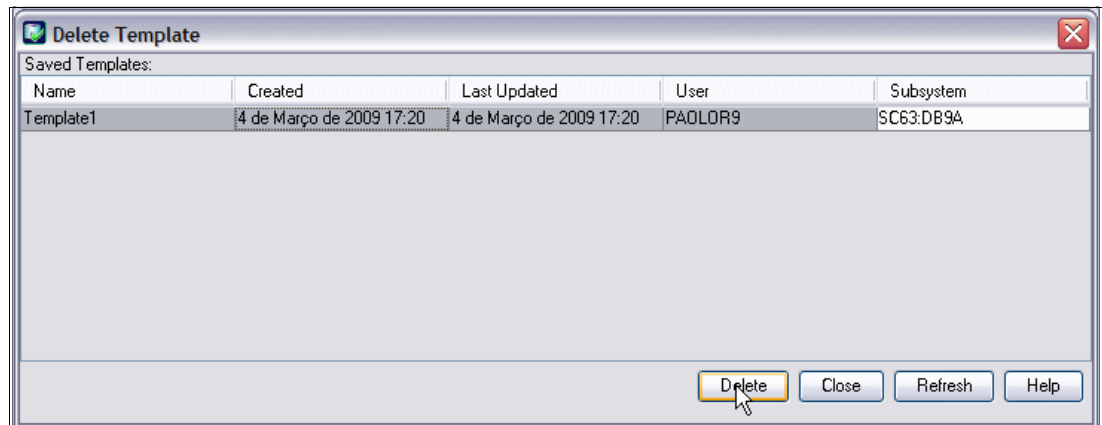


Figure 10-64 Delete Template

Notes:

- ▶ Log Analysis data (templates and jobs) are saved on the Audit Management Expert server. This enables you to access the data from any machine where you have successfully logged in to the Audit Management Expert server using the reporting interface.
- ▶ The output for Log Analysis Jobs is not saved on the Audit Management Expert server, by default it is stored in the local file system where the agent that submitted the Log Analysis job resides. This output can be retrieved and displayed in the reporting client.
- ▶ If there are EDITPROC edit routines defined on any tables for which Log Analysis is to be run, your product administrator must define the LOADLIB containing those EDITPROC programs in the product control file.

Log Analysis does not support tables that have been dropped, or dropped and re-created.



Audit Management Expert administration

In this chapter we provide recommendations for administering the product. We describe the separation of roles, performance, data collection and provide repository planning considerations.

This chapter contains the following:

- ▶ Separation of roles
- ▶ Control (DBA versus auditor)
- ▶ Performance monitoring
- ▶ Repository administration

11.1 Separation of roles

Separation of roles, or segregation of duties, has always been a major challenge to the auditing process. In general, auditors usually depend on developers or database administrators (DBAs) to collect and report information. The most critical drawback of this approach pertains to the integrity of the audited information.

DB2 Audit Management Expert provides separation of roles, resulting in data integrity, and more accurate reports. This frees up DBAs to perform DBA duties and allows auditors to run audit reports independently of the DBAs, resulting in easier, more accurate audits.

The DB2 Audit Management Expert administrator can specify how much visibility each auditor has to the auditable objects. Auditors are no longer required to issue DB2 commands to control the DB2 trace, as it is controlled within the product.

Auditors now have the ability to adhere to published industry standards and external auditing without relying on personnel who need to be monitored.

11.2 Control (DBA versus auditor)

DB2 Audit Management Expert is well-suited to enforce controls that govern DBAs, and to report on their activity. DBAs are trusted with sensitive data to do their jobs. They need to maintain sensitive data, copy the data, recover data, and load and reorganize it, to name some of their responsibilities. The continuous, automated auditing provided by DB2 Audit Management Expert removes the opportunity and the temptation to alter or omit, if authority permits, important data from the audit reports. Thus, the independence of the audit mechanism from personal involvement provides assurance that data in the reports has not been modified, and consequently, the accuracy of data and reports is more reliable. The DBA or DB2 system programmer may help the auditor during the install and initial configuration but then their user ID should be revoked so they do not have access to DB2 Audit Management Expert or critical audit data.

Important: The DB2 audit trace does not audit utilities such as COPY, RECOVER, REPAIR, DSN1COPY, DSN1CHKR, and DSN1PRNT. These utilities should be restricted and subject to well-defined controls

11.3 Performance monitoring

From the performance perspective, there are a few considerations to take into account when performing the tasks detailed in the sections that follow.

11.3.1 How to collect audit data

Using IFI only

When DB2 Audit Management Expert is configured to use IFI-based collection only, every table to be audited must be created or altered using AUDIT ALL, to start collecting audit trace information.

To alter table(s) as AUDIT ALL, there are two steps:

1. Issue an ALTER TABLE ... AUDIT ALL against a table or a set of tables to be audited. DB2 Audit Management Expert will not ALTER a table dynamically.

Usually, the DBA issues the ALTER for the auditor. Allowing this attribute, AUDIT ALL, against a table has no performance impact at all; the overhead of auditing a table does not come into play until the audit trace is activated.

Note: Audit trace records are not collected in SMF so there is no trace impact. Auditors do not need access to SMF data directly for DB2 Audit Management Expert audit information. DB2 Audit Management Expert Version 2.1 will only start the necessary traces according to the collection profile settings.

While audit traces classes are turned on, unwanted information is filtered out and not stored in the repository. The performance impact is directly related to the number of tables that are audited, and the number of transactions that access them.

2. Issue a rebind. ALTER TABLE ... AUDIT ALL invalidates all plans and packages that access this table, so those plans and packages are rebound. Also, AUTOBIND may or may not be turned on, presenting another performance consideration. In any event, a rebind of these affected plans and packages would circumvent the AUTOBIND issue and is advisable.

Recommendation: Identify and rebind all affected plans and packages once the ALTER ... AUDIT ALL completes.

Using IFI collection and ASC collection

For ASC-based collection, it is not necessary to set the AUDIT setting. The Audit SQL collector uses a collector developed in the IBM DB2 Query Monitor for z/OS. It is not necessary to have Query Monitor installed, but if you do, Query Monitor and the ASC component of DB2 Audit Management Expert for z/OS will use a shared master address space (shared collector) so that when both are running, the data is only collected once. If Query Monitor is not running, the DB2 Audit Management Expert ASC component starts the master address space. If the DB2 Audit Management Expert ASC component finds the master address space, it uses the master address space started by Query Monitor instead. If using both products, the overhead is significantly reduced by using the shared address space.

11.3.2 Controlling data collection

Deciding what data to collect

As mentioned above, the performance impact of auditing is directly dependent on the amount of audit data generated. The collected auditing information is written to external online performance buffers and then loaded into the repository by the agent. A DB2 subsystem can process a huge amount of data. If DB2 Audit Management Expert for z/OS was configured to capture all of that activity, it would incur unnecessary overhead, require a huge repository, and most likely, not all of the captured activity would be useful.

Recommendation: Audit only those tables and applications that are necessary. This minimizes the number of records collected. If the auditor is not interested in specific applications, those applications should be excluded from the collection profile.

Using filters to reduce the audit data to a useful subset

Filtering capability is available on both the collection side (before the data has been written to the repository), and on the reporting side (after the data has been collected and stored in the repository). It is wise to filter on the collection side instead of the reporting side to ensure that unnecessary data is not written to the repository.

The DB2 Audit Management Expert for z/OS administrator controls the amount of data collected and stored in its audit repository using a collection profile. With this collection profile you can collect a subset of the audit activity by filtering for any of the following items:

- ▶ Time
- ▶ General Audits: All failed authorizations, successful and failed authid changes, grants and revokes, IBM DB2 utilities, DB2 commands
- ▶ Reads
- ▶ Changes
- ▶ AUTHIDs
- ▶ Tables
- ▶ Plans
- ▶ WorkstationName
- ▶ WorkstationTrans

Using large number of includes and excludes

A major performance advantage of DB2 Audit Management Expert is its support of includes and excludes. For example, if we are sure that package A accesses table B securely, we may want to exclude that plan from the collection profile. The input/output (I/O) to the repository will be correspondingly reduced, and the overall performance of DB2 Audit Management Expert improved. Consider when there are a million accesses and a large number of includes and excludes, in this case, saving the I/O to the repository is extremely beneficial.

Including and excluding will increase CPU usage slightly, but from initial performance tests, the CPU usage of the DB2 Audit Management Expert agent was a small percentage of the total processing. Ultimately, it is more efficient to use CPU filtering to exclude an unwanted event instead of inserting it into the repository.

11.4 Repository administration

DB2 Audit Management Expert puts the audited data in its repository. From a repository administration perspective, there are a few considerations to take into account:

- ▶ The repository should be located in a production DB2 subsystem. Ideally, it should be separate from the monitored production subsystems and connected by fast network links.
- ▶ The repository table spaces should have regular RUNSTATS, REORG, and backups run like any production data.
- ▶ The default DDL to create the repository puts tables and indexes in Buffer Pool 0 (BP0) and should be reviewed for what is best in your environment.
- ▶ Repository data grows quickly, so it is important to establish a plan for archiving the data. Optim Data Growth for z/OS (see 6.2, “IBM Optim Data Growth Solution for z/OS” on page 111) can be used to archive data that needs to be kept to satisfy the regulatory compliance rules.
- ▶ The fast-growing tables are ADHEVENT, ADHEVENT_HOSTVS and ADH_SUMMARYUPDATE. Consider using compression on these tables.

Data Encryption for IMS and DB2 Databases Tool

IBM Data Encryption for IMS and DB2 Databases Tool runs as an exit or EDITPROC to encrypt data for storage and decrypt data for application use, protecting sensitive data residing on various storage media. Without this product, you need to write and maintain your own encryption software to use with such exits or within your applications. In this part we provide details on the use of Data Encryption for IMS and DB2 Databases Tool for encrypting DB2 for z/OS data.

This part contains the following chapters:

- ▶ Chapter 12, “Architecture and ICSF key management” on page 277
- ▶ Chapter 13, “Data Encryption tool installation and customization” on page 299
- ▶ Chapter 14, “Data encryption scenarios” on page 315
- ▶ Chapter 15, “Administration of encrypted objects” on page 345



Architecture and ICSF key management

In this chapter we discuss the following topics:

- ▶ Integrated Cryptographic Service Facility

This section provides an introduction of the z/OS architecture under which Data Encryption for IMS and DB2 Databases Tool executes.

- ▶ CEX2C configuration (HMC)

This section provides an introduction to the CEX2C hardware configuration and the use of PPINIT to create a DES Master Key to load and initialize the CEX2C. The discussion is intended to provide the DBA some background and sample window captures so he can approach his z/OS system programmer and describe the hardware and HMC configuration requirements

- ▶ DES master key generation

This is also intended to illustrate the use of the ICSF utility PPINIT, which will provide an expedient technique to generate DES Master keys and get a working ICSF environment which can support the implementation of IBM Data Encryption for IMS and DB2 Databases Tool.

12.1 Integrated Cryptographic Service Facility

Integrated Cryptographic Service Facility (ICSF) provides the cryptographic framework on z/OS under which products such as the Data Encryption for IMS and DB2 Databases Tool can perform encryption and decryption services. ICSF is delivered as part of the z/OS operating system, and provides facilities to assist in the following services:

- ▶ Key generation and distribution
 - Public and private keys
 - Secure and clear keys
 - Master keys
 - Reference to keys stored in CKDS through keylabels
- ▶ Access controls to ICSF services enforced with RACF
 - Controls to different services through CSFSERV
 - Access control applied to individual keys through CSFKEY
- ▶ Implementation of AES and TDES
- ▶ Software API to interact with cryptographic hardware

Key generation and distribution

ICSF provides a facility for security personnel, known as *key officers*, to create and administer various types of cryptographic keys. Of direct interest to the Data Encryption for IMS and DB2 Databases Tool is the ability to create data (both clear and secure) along with associated key labels and store these keys in a special VSAM data set called the Cryptographic Key Data Set (CKDS). Keys stored in the CKDS, and a special in-storage copy of the CKDS, are accessed for use by the Data Encryption for IMS and DB2 Databases Tool through the use of a key label. Key labels can be thought of as an index value that can be used to find a data encrypting key stored in the CKDS. When the Data Encryption for IMS and DB2 Databases Tool is used to generate a DB2 EDITPROC to perform DB2 encryption, the key label is provided to the Data Encryption for IMS and DB2 Databases Tool administrator by the key officer.

You can use either the Key Generator Utility Program (KGUP) or the ICSF key management APIs to generate and enter keys into the CKDS, or to maintain keys already existing in the CKDS. The keys are stored in records. KGUP can be either directly executed through user-coded JCL, or can be invoked by using the ICSF administration dialogs through ISPF panels. A record exists for each key that is stored in the CKDS. A record in the CKDS is called a key entry and has a label associated with it. When you call some ICSF callable services, you specify a key label as a parameter to identify the key for the callable service to use. Use KGUP to change the key value of an entry, rename entry labels, and delete entries in the CKDS.

While not appropriate for the discussion of the Data Encryption for IMS and DB2 Databases Tool and ICSF, there are other types of key generation that can be performed by ICSF, for example the generation of public/private key pairs to participate in secure sharing of encrypted data. An example of a product that uses this form of key implementation is the z/OS Encryption Facility.

When generating data encrypting keys to be stored in the CKDS, the key officer chooses the form of the key, either secure or clear key. The choice of secure key (see Figure 12-1 on page 279) will result in a data encrypting key being itself encrypted (or protected) with the DES or AES master key inside the CEX2C hardware feature prior to being stored in the CKDS. In addition, any subsequent access to this key is performed within the CEX2C, and the use of the decrypted data encrypting key as input to the encryption or decryption operation

also occurs completely within the CEX2C. This ensures that at no time is the value of the unencrypted data encrypting key exposed inside of operating system storage. The choice of secure or clear key needs to be communicated to the Data Encryption for IMS and DB2 Databases Tool administrator as the form of EDITPROC that is prepared will be dependent on the choice of secure or clear key made by the key officer.

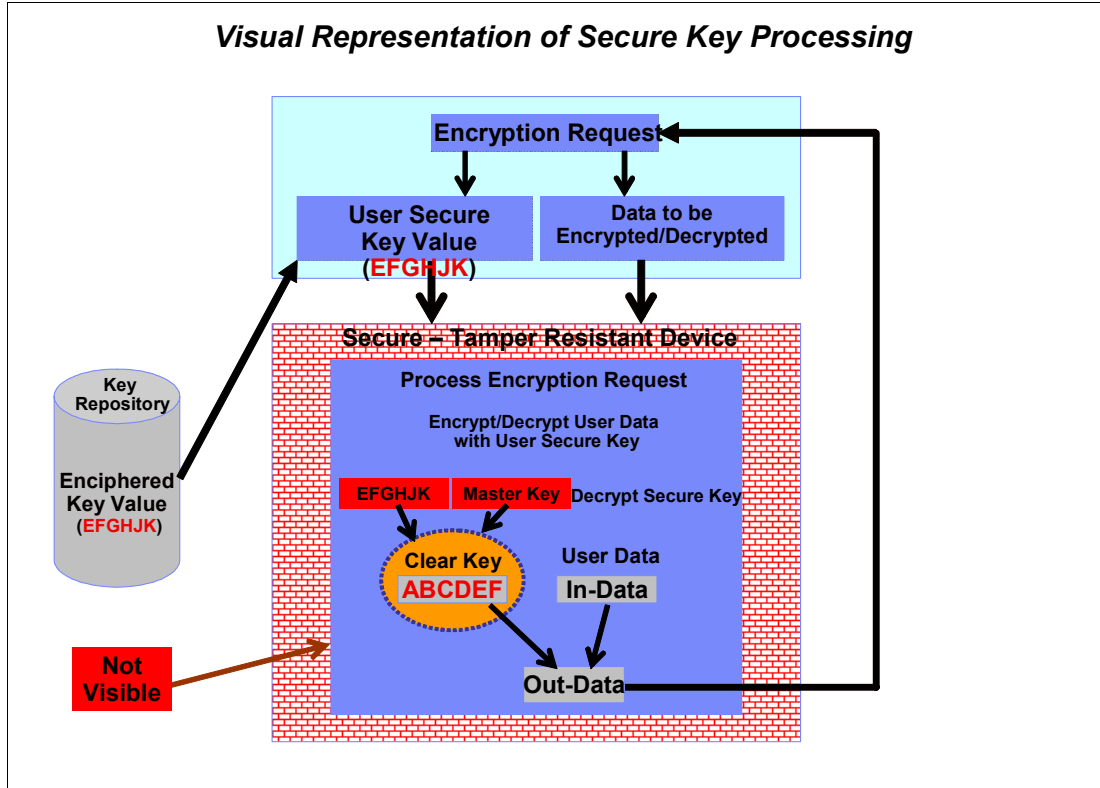


Figure 12-1 Visual representation of secure key

Contrasted with secure key processing, clear keys are stored inside the CKDS in clear-text format. A clear key encryption request will not run inside the CEX2C, rather the key is extracted directly from the CKDS as identified by the key label. The data encrypting key is used by the CPACF and clear key encryption and decryption is performed on the CPACF. See Figure 12-2.

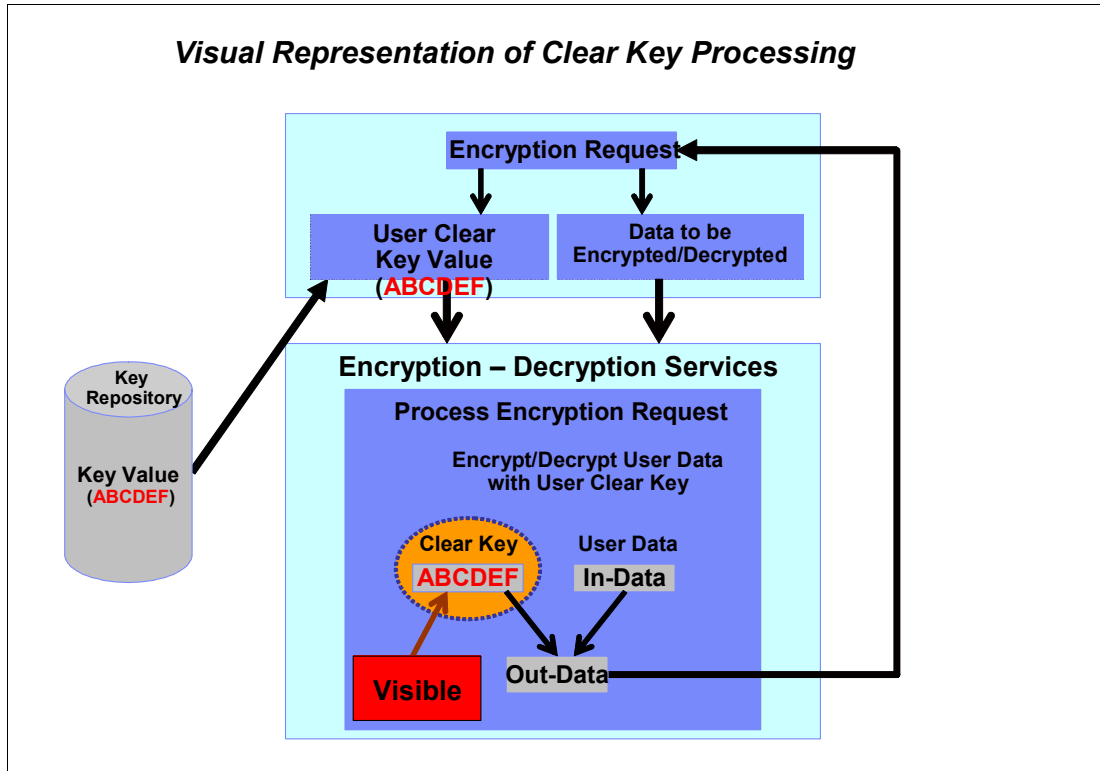


Figure 12-2 Visual representation of clear key

RACF controls for ICSF services

To secure access to different types of ICSF services, the security administrator can protect these services through the use of special RACF resource class definitions. You can use z/OS Security Server RACF to control which applications can use specific keys and services. This can help you ensure that keys and services are used only by authorized users and jobs.

You can also use RACF to audit the use of keys and services. The XCSFKEY class controls who can export a token using the Symmetric Key Export callable service (CSNDSYX). To set up these controls, you create and maintain RACF general resource profiles in the CSFKEYS class, the CSFSERV class and the XFACILIT class. The CSFKEYS class controls access to cryptographic keys with the key label, the CSFSERV class controls access to ICSF services, and resources in the XFACILIT class define a key store policy that controls the use of key tokens that are stored in the CKDS and PKDS. For more information, refer to *z/OS Cryptographic Services Integrated Cryptographic Service Facility Administrator's Guide*, SA22-7521-13.

Additionally, one can elect to further control access to information about the ICSF environment by restricting access to the ICSF dialog ISPF panels.

Implementation of AES and TDES

While the particulars of which specific form of AES and TDES encryption are dependent on the type of System z processor being used, the Data Encryption for IMS and DB2 Databases Tool can support both AES and TDES encryption.

Typically, DES/TDES key lengths are referred to as 8-byte, 16-byte, or 24-byte. In all cases, one bit is used for parity, so for an 8-byte key (64-bits) the key is really only 56-bits. In any case, TDES can use a double length (16-byte) or triple length (24-byte) key. Both DES and TDES are well supported on the z9 and z10.

The z10 provides hardware support for both 192-bit and 256-bit AES encryption. This enables support for CPACF with the KMC hardware instruction. The z9 provides hardware support for 128-bit AES encryption only, AES 192-bit and 256-bit encryption support on z9 is only implemented through software. In general, the use of software implemented encryption is discouraged as the performance characteristics are generally poor.

One other characteristic is the relationship between CPACF and the CP engine in a System z CEC. On the z9, there is a one-for-one relationship between CPACF and CP processor. In the z10, there is one CPACF shared between two CP processors.

Unlike the choice of clear or secure key, the EDITPROC prepared by the Data Encryption for IMS and DB2 Databases Tool administrator does not need to be aware of the type of encryption algorithm being specified. This is controlled by the key officer during the generation of the user key through KGUP specification.

12.2 CEX2C configuration (HMC)

The intent of this section is not to describe the details on how you activate CEX2C features on a System z processor, but to give some background to enable the DB2 professional to guide the z/OS system programmer in this exercise.

In many situations, there will be applications that already exploit the cryptographic features supported by the CEX2C coprocessor. In this discussion, we assume that a CEX2C feature is installed into the system, but is undefined to the LPAR where DB2 runs.

LPAR mode and domains

A CEX2C coprocessor or accelerator can be shared by up to 16 LPARs. This implies that if more than 16 active LPARs are intended to share the CEX2C, more than one card is needed.

The CEX2 coprocessors or accelerators can be shared between logical partitions. What is actually shared are the cryptographic engines. The master key cannot be shared, as it remains the secret of an installation/logical partition. The CEX2C coprocessor has 16 physical domains. Each domain is a separate physical set of registers to hold a master key. Each logical partition is given a domain, through the image profile, and only one active partition can access a specific domain in a specific coprocessor at any point in time. The HMC (Hardware Master Console) is used by the z/OS system programmer to manage image profiles.

Figure 12-3 shows the relationship between LPAR and Domain. Note that each domain has its own unique master key loaded into the CEX2C registers.

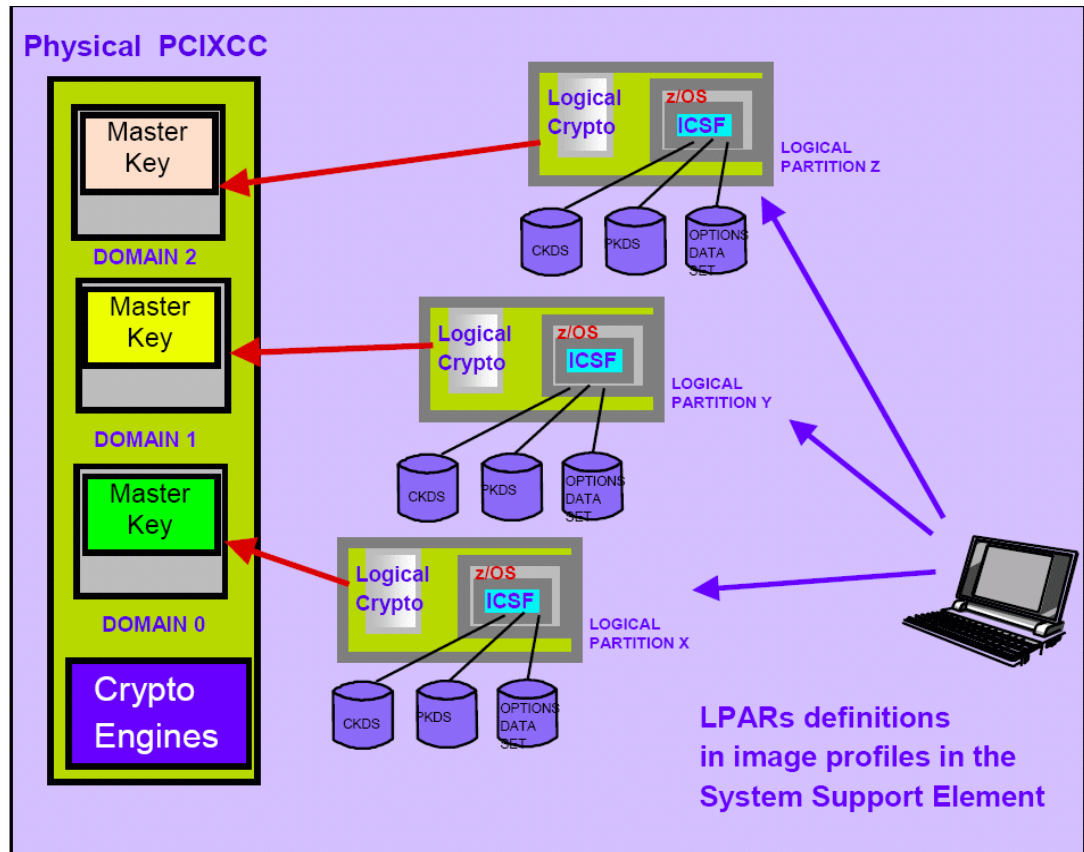


Figure 12-3 LPAR and domain relationship

The DES master key is installed by the security officers using PPINIT or Master Key Management ISPF dialog. The security officers install the master key for each partition. The master key is set in the logical partition domain and remains isolated from the other partitions. Each domain will have its own master key loaded into the CEX2C registers. Although the same key can be used in different domains, in the case of a parallel sysplex environment where DB2 data sharing is supported, this is a requirement. The TSO/E ISPF panels used to manage a secure coprocessor actually manage the partition's domain.

The z/OS system programmer uses the Hardware Management Console or the System Support Element to maintain LPAR definitions in image profiles. The image profile is where the domain assignments are made.

Making CEX2C features available to ICSF

On our environment, we start out with no CEX2C features available. In this scenario, when the ICSF-started task comes up, you will see the following message in the ICSF started task job log as illustrated in Figure 12-4.

```
STC22210 CSFM506I CRYPTOGRAPHY - THERE IS NO ACCESS TO ANY CRYPTOGRAPHIC COPROCESSORS
STC22210 CSFM100E CRYPTOGRAPHIC KEY DATA SET, PAOLR5.SC63.CSFCKDS IS NOT INITIALIZED
STC22210 CSFM001I ICSF INITIALIZATION COMPLETE
STC22210 CSFM126I CRYPTOGRAPHY - FULL CPU-BASED SERVICES ARE AVAILABLE
```

Figure 12-4 ICSF startup without CEX2C available

As indicated in the message log, there are no cryptographic coprocessors allocated to this LPAR, and we have an un-initialized CKDS, which we expect as we need CEX2C resources to initialize the CKDS. Note that there are some limited cryptographic services available, but for our purposes, these will not allow the IBM Data Encryption for IMS and DB2 Databases Tool to perform the necessary crypto work.

Important: Remember, in this scenario we are showing how to initialize and run in a CEX2C-enabled environment. If your encryption requirements are clear keys only, and you are running with the ICSF version HCR7751, it is possible to initialize a CKDS and generate clear keys without CEX2C hardware being available.

Once ICSF comes up, we can also look at the ICSF panel for Coprocessor Management, which will show no coprocessors available. We see an example of this in Figure 12-5. We will see after the ICSF feature assignment in the HMC, that this display will change.

```
----- ICSF Coprocessor Management -----
COMMAND ===>                                SCROLL ===>

Select the coprocessors to be processed and press ENTER.
Action characters are: A, D, E, K, R and S. See the help panel for details.

  COPROCESSOR      SERIAL NUMBER    STATUS
  -----          -
***** Bottom of data *****

F1=HELP      F2=SPLIT      F3=END      F4=RETURN      F5=RFIND      F6=RCHANGE
F7=UP        F8=DOWN       F9=SWAP     F10=LEFT      F11=RIGHT     F12=RETRIEVE
```

Figure 12-5 Coprocessor management panel with no assigned CEX2C

Hardware Management Console operation overview

From the system support element, the z/OS system programmer should select the Hardware Management Console (HMC) Console Workplace and designate the Groups view, as shown in Figure 12-6.

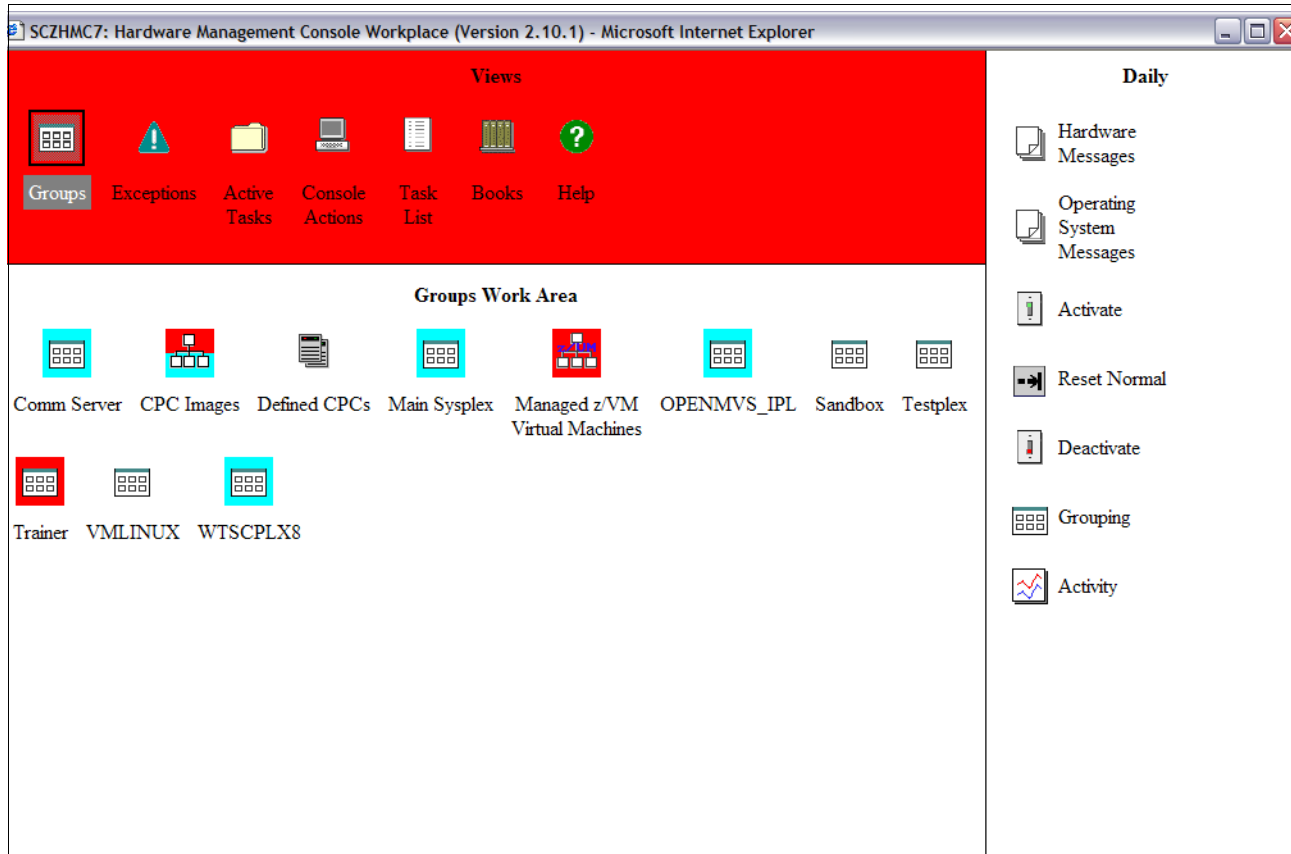


Figure 12-6 HMC Console workplace

The CPC images work area workplace is shown next. There will probably be several images displayed on your workarea, as previously defined by the z/OS system programmer. Images can be equated to LPAR definitions. In our case, the LPAR where we will be conducting our cryptographic configuration is SANDBOX:SC63.

This is shown in Figure 12-7 on page 285. Also highlight the **customize activation profiles** button.

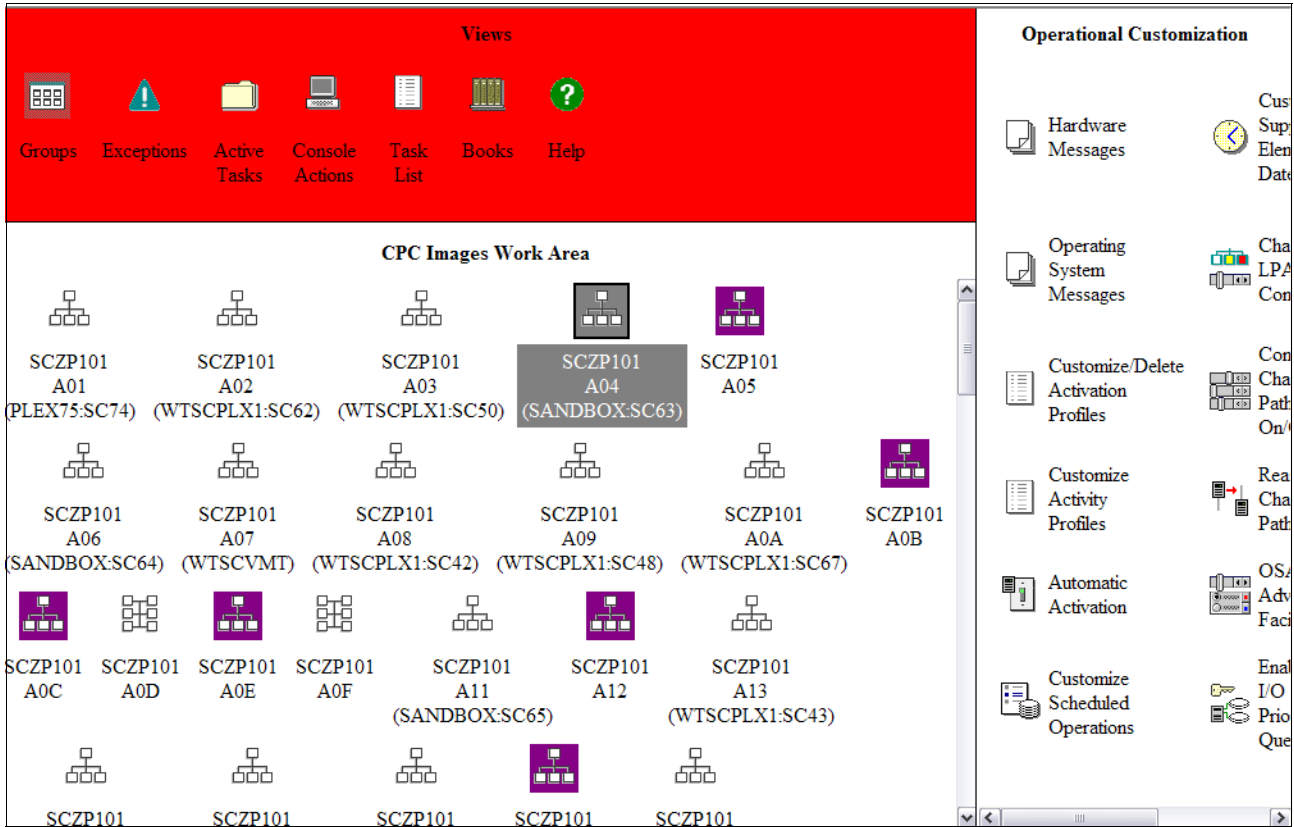


Figure 12-7 CPC Images Workarea - LPAR Designation

Once the selected LPAR profile has been selected, the next window is the activations profile list. This is where the choice of profile is made. In our case, we select the default image A04. This is shown in Figure 12-8.

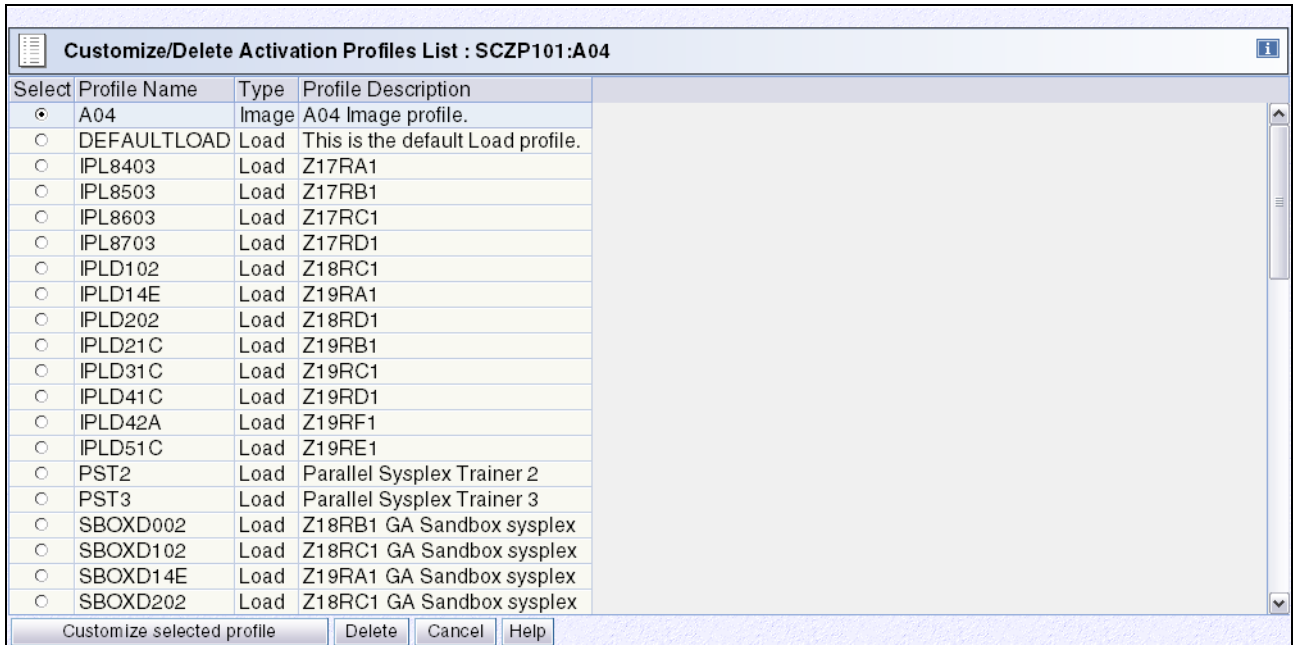


Figure 12-8 Activation profiles list

From the profiles list, we navigate to the control domain/usage domain workspace. This is where the list of eligible crypto features are displayed and associated with each domain. In our case, there are no CEX2C features assigned to this domain. This is designated by the lack of check boxes selected on the right side of the workspace, as shown in Figure 12-9.

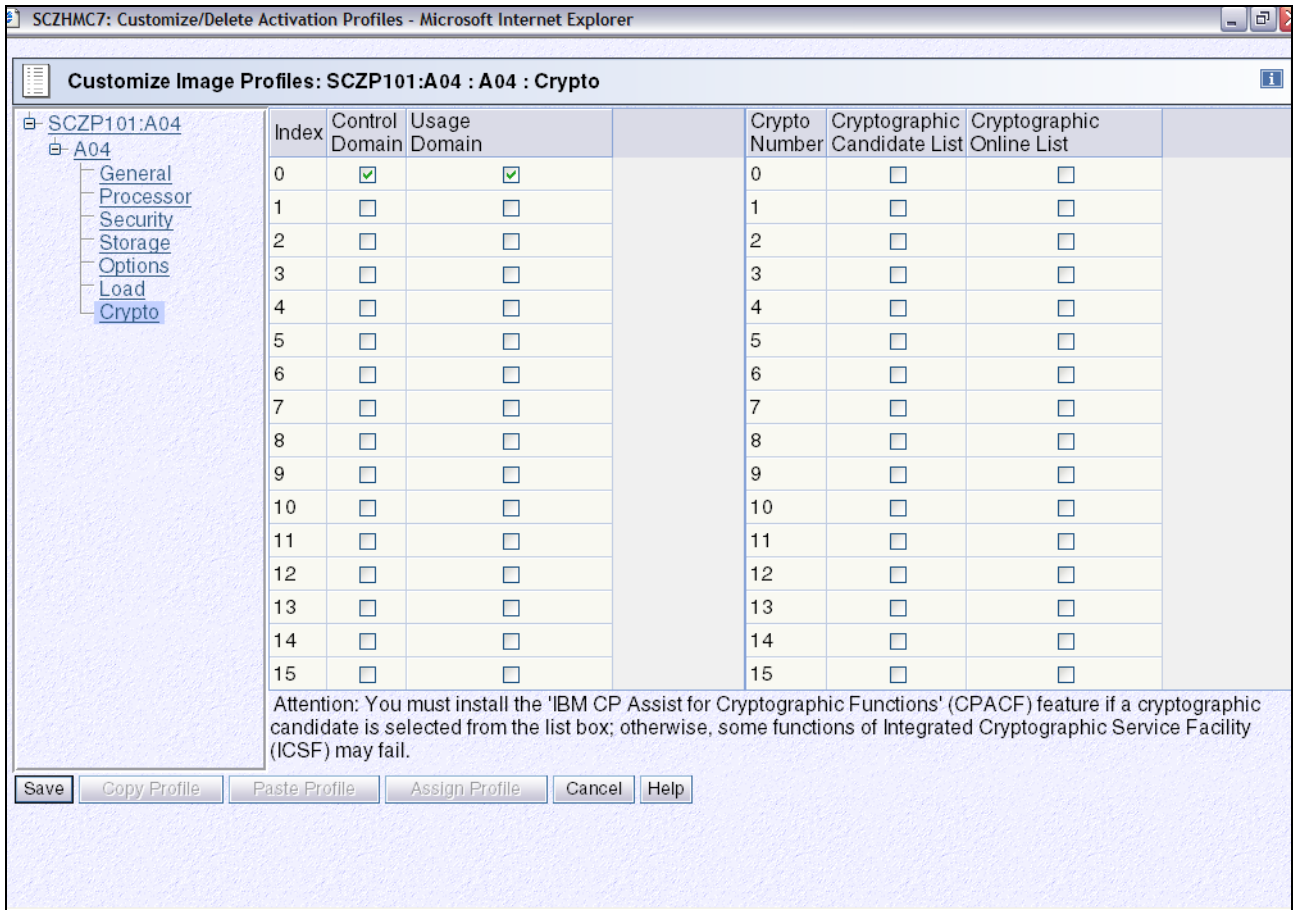


Figure 12-9 Control Domain/Usage Domain Assignment

In our case, we are going to assign Domain #2 on Crypto engine number 1 to LPAR 2, as shown in Figure 12-10 on page 287.

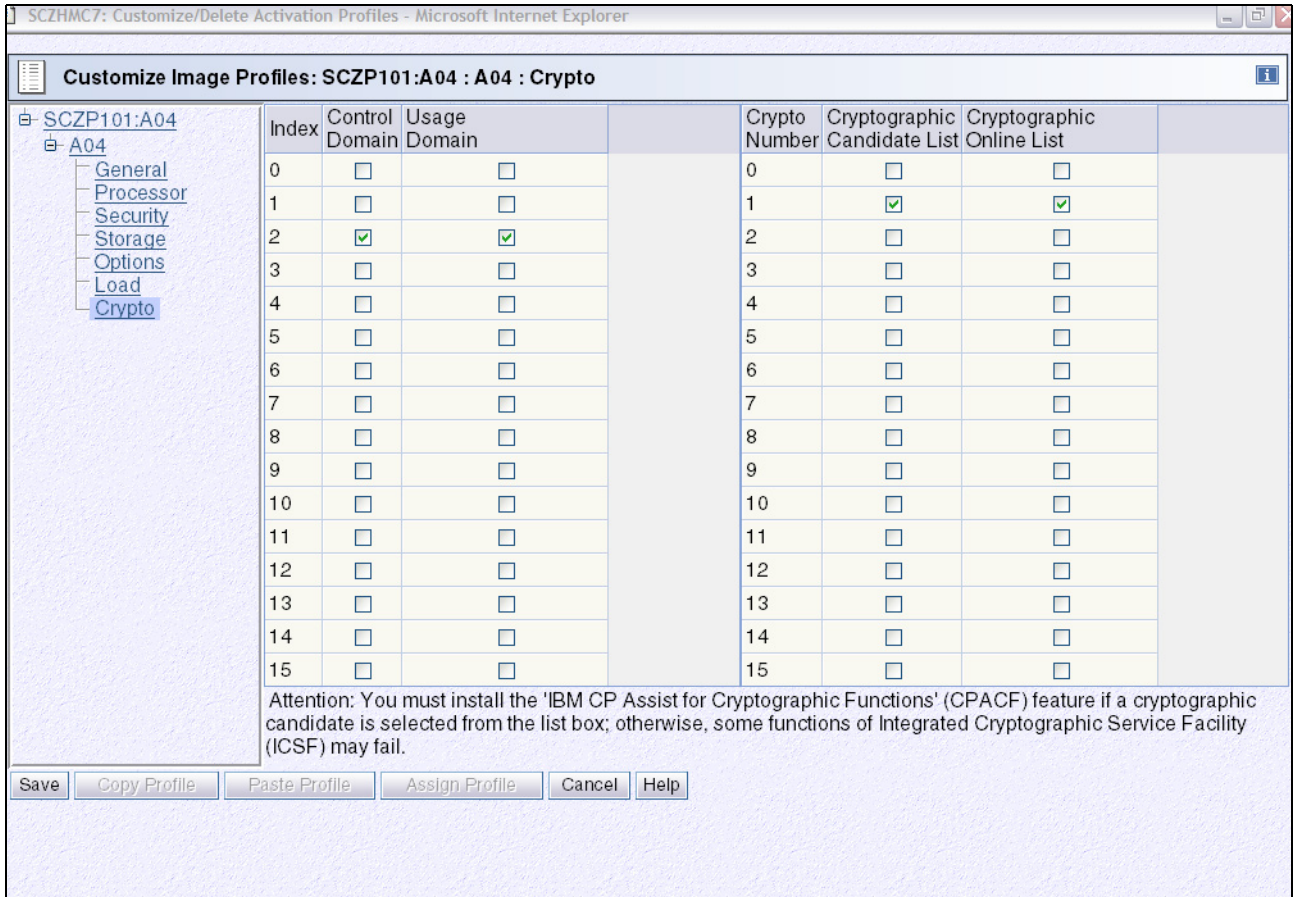


Figure 12-10 Control Domain Cryptographic Candidate designation

Once this has been accomplished, the z/OS system programmer will need to schedule a domain inactivation/reactivation. On a z10, you can use the HMC/SE¹ to activate the changes. At that point, the CEX2C feature should be available to the LPAR. Using the ICSF administration panel, this can be confirmed as shown in Figure 12-11 on page 288. When the CEX2C feature is installed and available, the status changes to ONLINE. At this point, the CEX2C needs to have DES master keys loaded either through the ICSF Master Key Management window, or through the use of PPINIT.

¹ Hardware Management Console (HMC) and Support Element (SE)

```

----- ICSF - Coprocessor Hardware Status -----
COMMAND ==>                                SCROLL ==>
                                           CRYPTO DOMAIN: 2

REGISTER STATUS                            COPROCESSOR E01

Crypto Serial Number      : 94000264
Status                    : ONLINE
DES Master Key
  New Master Key register : EMPTY
  Verification pattern    :
  Hash pattern            :
                          :
  Old Master Key register : EMPTY
  Verification pattern    :
  Hash pattern            :
                          :
  Current Master Key register : EMPTY
  Verification pattern    :
  Hash pattern            :

Press ENTER to refresh the hardware status display.
Press END   to exit to the previous menu.

```

Figure 12-11 ICSF coprocessor hardware status

12.3 DES master key generation

Once we have ensured that one or more CEX2C coprocessor features are online and accessible, we will describe how to get an initial set of DES master key values loaded into the coprocessor registers.

12.3.1 Loading cryptographic processors with DES master key

Master keys are used to protect sensitive cryptographic keys that are active on your system. The number and types of master keys you need to enter depends on your hardware configuration and application requirements.

For example:

- ▶ On the CEX2C, the DES master key (DES-MK) protects DES secure keys and the asymmetric-keys master key (ASYM-MK) protects RSA keys.
- ▶ On the CEX2C, the AES secure key DES-MK protects DES secure keys and an AES-MK protects AES secure keys. AES secure key is supported on the z9 and the z10 with HCR7751.

When you start ICSF for the first time on your system, you have the option to enter master keys and initialize the cryptographic key data set (CKDS) and PKA cryptographic key data set (PKDS). You can generate and enter the keys you use to perform cryptographic functions. The master keys you enter can protect secure keys stored in the CKDS and the PKDS.

Attention: For our scenarios, we will not be using any crypto functions that require the use of PKA services, so we will not discuss loading PKA master keys (ASYM-MK). Only the DES Master key will be used.

All DES and AES keys, except the master keys, can be stored in the CKDS. There are several methods you can use to enter keys into the CKDS:

- ▶ Key generator utility program (KGUP)
Use KGUP to enter keys into the CKDS. This option can be used on any processor or server model. We will illustrate this in our scenario.
- ▶ CKDS key management APIs
Program applications to use key management APIs to enter keys into the CKDS. This option can be used on any processor or server model.

We demonstrate the use of KGUP to create clear keys to be used by the Data Encryption for IMS and DB2 Databases Tool in Chapter 13, “Data Encryption tool installation and customization” on page 299. Even when using clear keys, ICSF requires that there be a CEX2C feature active, loaded with keys, and the CKDS initialization needs to be performed.

Attention: When running ICSF at the HCR7751 level, if you are using clear keys only. It is possible to initialize and store clear keys in a CKDS without the CEX2C features being available. We will discuss this further in this chapter.

CKDS Allocation

We need to create and initialize the CKDS and enter the master keys. The CKDS is a VSAM Key Sequenced Data. It has the following attributes, and can be initialized as shown in Figure 12-12.

```
DELETE ('PAOLOR5.SC63.CSFCKDS') CL PURGE
DEFINE CLUSTER (NAME(PAOLOR5.SC63.CSFCKDS) -
              RECORDS(100 50)           -
              RECORDSIZE(252,252)      -
              KEYS(72 0)                -
              FREESPACE(10,10)         -
              VOLUME(SBOX20)           -
              SHAREOPTIONS(2)          -
              DATA (NAME(PAOLOR5.SC63.CSFCKDS.DATA) -
              BUFFERSPACE(100000)      -
              ERASE                      -
              WRITECHECK)              -
              INDEX (NAME(PAOLOR5.SC63.CSFCKDS.INDEX))
```

Figure 12-12 IDCAMS VSAM Define statements for CKDS

We recommend that this data set be placed on a volume that is backed up on a frequent basis, and will be available at disaster recovery recovery site. Many customers treat the CKDS as a system critical data set, and place it on the same volume as their SYSRES and other system critical data sets. Ensure that the CKDS is not eligible for HSM migration and other such disruptive events, and placing the CKDS a mirrored volume is also recommended. Consider protecting the CKDS with a RACF UACC(NONE), then only allowing ICSF key officers and the ICSF started task access.

ICSF parameters

After the CKDS is allocated, you should ensure that ICSF points to the newly allocated CKDS to allow for initialization and master key creation. The ICSF address space (in our environment we call this address CSF) is controlled by a parameter member stored in SYS1.PARMLIB, and the ICSF started task JCL, which is typically stored in SYS1.PROCLIB.

In our environment, the sample JCL used for starting the ICSF task is shown in Figure 12-13. We specify the use of a symbolic for SUFFIX, this allows us to start different forms of the ICSF started task by using the appropriate value for SUFFIX in the z/OS START command

```
//CSF PROC SUFFIX=00
// EXEC PGM=CSFMMAIN,REGION=0M,TIME=1440
//CSFLIST DD SYSOUT=*,DCB=(LRECL=132,BLKSIZE=132),HOLD=YES
//CSFPARM DD DSN=SYS1.PARMLIB(CSFPRM&SUFFIX),DISP=SHR
```

Figure 12-13 ICSF JCL procedure

When we start the ICSF, or in our case CSF, we use the following z/OS console command to start the started task, as shown in Figure 12-14. For our example, we have a member named CSFPRMEM in SYS1.PARMLIB. This contains the various ICSF initialization parameters discussed in more detail below.

```
S CSF,SUFFIX=EM
```

Figure 12-14 ICSF z/OS Start Command

ICSF initialization parameters control the execution mode of ICSF, including the names of the CKDS/PKDS that are dynamically allocated at ICSF startup. Figure 12-15 shows how we used the SYSCON method of substituting the SYSNAME. This allows for the sharing of ICSF startup JCL across multiple LPARs.

```
CKDSN(PAOLR5.&SYSNAME..CSFCKDS)
PKDSN(PAOLR5.&SYSNAME..CSFPKDS)
COMPAT(NO)
SSM(YES)
KEYAUTH(NO)
CKTAUTH(YES)
CHECKAUTH(NO)
TRACEENTRY(1000)
USERPARM(USERPARM)
COMPENC(DES)
REASONCODES(ICSF)
PKDSCACHE(64)
```

Figure 12-15 ICSF startup parameters

We have taken most of the defaults with a couple of exceptions. We discuss all of the startup parameters as follows:

- ▶ CKDSN

Data set name of the CKDS

- ▶ PKDS

Data set name of the PKDS

- ▶ CHECKAUTH

Indicates whether ICSF performs security access control checking of Supervisor State and System Key callers. If you specify CHECKAUTH(YES), ICSF issues RACROUTE calls to perform the security access control checking and the results are logged in RACF SMF records that are cut by RACF. If you specify CHECKAUTH(NO), the authorization checks against resources in the CSFSERV class are not performed resulting in a significant performance enhancement for supervisor state and system key callers. However, the authorization checks are not logged in the RACF SMF records. We chose CHECKAUTH (NO).

Tip: In a DB2 environment, the RACROUTE check is performed with the AUTHID of the SQL requestor, because SQL runs under its owning TCB. With AUTHCHECK set to YES, each time an EDITPROC is loaded (the first invocation use of each clear key), ICSF will expect the user ID of the request to have authority to the CSFSERV class entry for that protected keylabel. It will be difficult to determine this, and you must connect all of the possible combination of SQL requestors to the CSFSERV group to pass the AUTHCHECK. Our recommendation is to set AUTHCHECK to NO, and provide additional protection by ensuring that linking into APF libraries such as SDSNEXIT is restricted. Problem state callers will always be subject to CSFSERV class checking.

- ▶ CKTAUTH

Decides if authentication will be performed for every CKDS record read from disk. We set this to YES, with our implementation the CKDS is only read when the in-storage copy of the CKDS is built, at start-up or when it is refreshed on command. If you implement a secure key environment, depending on the number of secure key requests, this might be a parameter with performance implications, as with secure key, the CKDS is read on each request, as no copy of the key is kept in storage, all operations occur inside the boundary of the CEX2C feature.

- ▶ COMPAT

Indicates whether ICSF runs in compatibility mode, non-compatibility mode, or coexistence mode with PCF. We are running with COMPAT set to NO, this is a feature that allowed for migration from earlier forms of encryption that were implemented on early generation processors.

- ▶ COMPENC

This keyword is no longer supported but is tolerated.

- ▶ KEYAUTH

Indicates whether or not ICSF authenticates a key entry after ICSF retrieves one from the in-storage CKDS. We specified NO. If you specify KEYAUTH(NO), ICSF does not perform this authentication and gains a small performance enhancement.

► SSM

Specifies whether or not an installation can enable special secure mode (SSM) while running ICSF. SSM allows you to enter clear keys and generate clear PINs. You need proper processes and procedures in place while doing so. You must enable SSM for KGUP to permit generation or entry of clear keys and to enable the secure key import or clear pin generate callable services. We specify SSM(YES), as we are using KGUP to generate clear keys for use by Data Encryption for IMS and DB2 Databases Tool.

For the purposes of our scenario, all other parameter were coded using the ICSF defaults and are uninteresting for this discussion.

CKDS initialization

Once allocated, we then need to initialize the CKDS to enter DES master keys and generate user keys through KGUP. Figure 12-18 on page 293 shows the ICSF main menu, all of the ICSF administration tasks, typically performed by key officers or security personnel, can be performed using the dialogs below.

12.3.2 PPINIT and CKDS initialization

Prior to using the CKDS, we must load a set of master key values into the CEX2C hardware registers. The CKDS initialization process will link the master keys in the hardware registers with the CKDS, by loading the hash and validation values associated with those keys into the header record of the CKDS. We already have assigned one or more CEX2C hardware elements to this LPAR domain. As we can see in Figure 12-16, for COPROCESSOR E01, there is a status of ONLINE. If you recall, from Figure 12-5 on page 283, there were no coprocessors displayed. Also of interest are the various master key register fields indicating a status of EMPTY. This is indicative that we need to load master key values in the registers.

```
----- ICSF - Coprocessor Hardware Status -----
COMMAND ===>                                     SCROLL ===>
                                                CRYPTO DOMAIN: 2

REGISTER STATUS                                COPROCESSOR E01
                                                More:      +
Crypto Serial Number      : 94000264
Status                    : ONLINE          <----- assigned but no
master key loaded
DES Master Key
  New Master Key register : EMPTY
  Verification pattern    :
  Hash pattern            :
                          :
  Old Master Key register : EMPTY
  Verification pattern    :
  Hash pattern            :
                          :
  Current Master Key register : EMPTY
  Verification pattern    :
  Hash pattern            :

Press ENTER to refresh the hardware status display.
Press END   to exit to the previous menu.
```

Figure 12-16 ICSF Coprocessor Hardware Status

When we start up ICSF, we show the status illustrated in Figure 12-17. Again, the crypto environment is not sufficiently configured for use by the IBM Data Encryption for IMS and DB2 Databases Tool.

```
CSFM124I CRYPTO EXPRESS2 COPROCESSOR E01, SERIAL NUMBER 94000264, NOT INITIALIZED
```

Figure 12-17 ICSF Startup messages with CEX2C available but no keys loaded

To load master keys into a CEX2C, there are only two options, one is to use the PPINIT utility to generate a Pass Phrase Master Key, or use the Master Key Management facility of the ICSF administration panels. The Master Key Management panels allows for the creation of a multiple part master key, and while is recommended for robust production implementations, it requires the availability of random number generation services, which require the use of service facilities supported by CEX2C. Obviously this would present an unresolved puzzle, so to provide a mechanism to help with the first time load of master keys, most customers will use the PPINIT Pass Phrase Master Key approach. This approach removes the requirement for a random number generation process as part of key preparation, and for the purposes of our scenarios, is the technique we have chosen to implement

Important: PPINIT is not intended for production environment. It is only recommended for the first time master key generation process. Once implemented, the recommendation is to perform a key rotation ceremony and replace the pass phrase master key with a multiple part master key prepared with Master Key Management facilities

Using the ICSF ISPF window, we navigate to the PPINIT panel, as seen in Figure 12-18.

```
HCR7751 ----- Integrated Cryptographic Service Facility-----
OPTION ==> 6
Enter the number of the desired option.

 1 COPROCESSOR MGMT - Management of Cryptographic Coprocessors
 2 MASTER KEY MGMT - Master key set or change, CKDS/PKDS Processing
 3 OPSTAT           - Installation options
 4 ADMINCNTL       - Administrative Control Functions
 5 UTILITY          - ICSF Utilities
 6 PPINIT          - Pass Phrase Master Key/CKDS Initialization
 7 TKE             - TKE Master and Operational Key processing
 8 KGUP            - Key Generator Utility processes
 9 UDX MGMT        - Management of User Defined Extensions

Licensed Materials - Property of IBM
5694-A01 Copyright IBM Corp. 1989, 2008. All rights reserved.
US Government Users Restricted Rights - Use, duplication or
disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Press ENTER to go to the selected option.
Press END to exit to the previous menu.
```

Figure 12-18 ICSF Main ISPF Menu

The PPINIT Utility generates master key values from a passphrase and loads those keys into the hardware registers.

Once a master key is loaded, you can perform the following tasks:

- ▶ Encipher and decipher data using an unencrypted key
- ▶ Generate a checksum, verification and hash pattern for a key part for key entry
- ▶ Generate, delete, import, and export key entries in the PKDS.

In our example we use this option to generate a passphrase-based master key. In our scenario, the following values are shown for illustrative purposes in Figure 12-19

▶ **PASSPHRASE**

The Passphrase is a character string between sixteen and sixty-four characters long. All characters in the EBCDIC character set are allowed. Embedded blanks are allowed. Leading and trailing blanks are truncated. For our master key, we chose to use the phrase 'Paolo Says There is No Free Lunch'. Once entered, this becomes the string on which the master key is generated. You must retain the EXACT value of the passphrase in case you need to reload the master key into your CEX2C hardware registers. This will include any scenario involving the use of the encoded CKDS at a disaster recovery exercise.

▶ **CKDS/PKDS**

The names of the CKDS and PKDS are the valid preallocated VSAM data sets.

- ▶ The Initialize System option allows you to load the DES and asymmetric master keys for all coprocessors and initialize the CKDS and PKDS. The CKDS and PKDS must be empty.
- ▶ Once you have an initial set of online coprocessors with valid master keys loaded, as you add new CEX2C features, you can use the **Add coprocessors** option to initialize these additional coprocessors with the same master key values. In our environment, we only have a single coprocessor allocated. In a typical production implementation, you would have a minimum of 2 coprocessors available to provide redundancy and failover support in the event of a hardware failure

```
----- ICSF - Pass Phrase MK/CKDS/PKDS Initialization -----
COMMAND ==>>

                                                                 More:  +
Enter your pass phrase (16 to 64 characters)
==>> PAOLO SAYS THERE IS NO FREE LUNCH

Select one of the initialization actions then press ENTER to process.

 2 Initialize system - Load the DES and asymmetric master keys to all
   coprocessors and initialize the CKDS and the PKDS.
   CKDS ==>> PAOLOR5.SC63.CSFCKDS
   PKDS ==>> PAOLOR5.SC63.CSFPKDS

_ Reinitialize system - Load the DES and asymmetric master keys to all
  coprocessors and make the specified CKDS and PKDS the current key data
  sets.
  CKDS ==>>
  PKDS ==>>

_ Add coprocessors - Initialize additional online coprocessors with the
  same DES and asymmetric master keys.
```

Figure 12-19 PPINIT Master Key and CKDS Initialization

Once PPINIT has been performed, there are master key values loaded in the CEX2C hardware registers, and the CKDS has verification and hash pattern values stored into the header (control) record. This information is used at each ICSF startup to ensure that the master key values in the hardware registers match with the master key used to initialize and encrypt keys on the CKDS. We demonstrate a failure scenario built around this master key mismatch in Part 6, “Appendixes” on page 365.

Once the master key has been loaded into the hardware register, if we use the ICSF Coprocessor Hardware Status panel, we see some changes in the values displayed, as shown in Figure 12-19 on page 294.

Notice that the status of the Crypto Hardware has been changed from ONLINE to ACTIVE. In addition, the Current Master Key register field has now been flagged as VALID. Finally, we see that the Current Master Key Verification and Hash pattern fields now contain values.

```

----- ICSF - Coprocessor Hardware Status -----
COMMAND ==>                                     SCROLL ==>
                                                CRYPTO DOMAIN: 2

REGISTER STATUS                                COPROCESSOR E01
                                                More:      +
Crypto Serial Number      : 94000264
Status                    : ACTIVE
DES Master Key
  New Master Key register  : EMPTY
  Verification pattern     :
  Hash pattern            :
                          :
  Old Master Key register  : EMPTY
  Verification pattern     :
  Hash pattern            :
                          :
  Current Master Key register : VALID
  Verification pattern     : F070C0451E30EC3F
  Hash pattern            : 2100222009E844DC

Press ENTER to refresh the hardware status display.
Press END   to exit to the previous menu.

```

Figure 12-20 Coprocessor Hardware status - After PPINIT

At this point, we have a fully configured CEX2C and CKDS which will support the use of the different cryptographic elements exploited by IBM Data Encryption for IMS and DB2 Databases Tool.

12.3.3 HCR7751 and CKDS operations without CEX2C

As mentioned in 7.1, “System z integrated cryptography” on page 128, for the customer who is running ICSF release HCR7751, and has no ICSF key requirements outside of supporting AES or TDES clear keys, that configuration can be supported without a CEX2C.

To support this configuration, there is a separate technique for initializing the CKDS, because without the CEX2C and associated master keys, we would not have the requirement for the master key hash and verification patterns to be stored in the CKDS header record. We will refer to this type of CKDS as an unprotected CKDS.

To support the generation of an unprotected CKDS, you will first create a CKDS using the IDCAMS example shown in Figure 12-12 on page 289. Once this is done, you will need to ensure that the CKDS is named in the ICSF startup parameter deck. This is described and shown in 290.

Starting with the ICSF menu shown in Figure 12-21, select and execute the INIT CKDS function.

```
----- ICSF - Master Key Management -----  
OPTION ==>  
  
Enter the number of the desired option.  
  
 1 INIT/REFRESH/UPDATE CKDS - Initialize a Cryptographic Key Data Set or  
                             activate an updated Cryptographic Key Data Set  
 2 SET MK                    - Set a symmetric (DES or AES) master key  
 3 REENCIPHER CKDS          - Reencipher the CKDS prior to changing a symmetric  
                             master key  
 4 CHANGE MK                 - Change a symmetric master key and activate the  
                             reenciphered CKDS  
 5 INITIALIZE PKDS           - Initialize or update a PKDS Cryptographic  
                             Key Data Set header record  
 6 REENCIPHER PKDS           - Reencipher the PKA Cryptographic Key Data Set  
 7 REFRESH PKDS              - Activate an updated PKA Cryptographic Key Data Set  
  
Press ENTER to go to the selected option.  
Press END  to exit to the previous menu.
```

Figure 12-21 Master Key Management menu

Once selected, we enter the CKDS named in our ICSF initialization parameters, Because we are operating in an environment without any CEX2C, set the parameter for MAC AUTHENTICATION to NO, as we do not have any hash or verification pattern data in the hardware registers (or for that matter, we do not have any secure hardware). Once entered, we see the resultant INITIALIZATION COMPLETE status message shown in Figure 12-22 on page 297


```

CSFCKD10 ----- ICSF - Initialize a CKDS  INITIALIZATION COMPLETE
COMMAND ==> 1

Enter the number of the desired option.

  1 Initialize an empty CKDS (creates the header and system keys)
    MAC record authentication required? (Y/N) ==>
  2 REFRESH - Activate an updated CKDS

Enter the name of the CKDS below.

CKDS ==> 'PAOLOR5.SC63.CSFCKDS'

Press ENTER to execute your option.
Press END  to exit to the previous menu.

```

Figure 12-22 CKDS Initialization - No CEX2C

Attention: During our scenario, we discovered that the IBM Data Encryption for IMS and DB2 Databases Tool was using a value in the CVT to determine the absence or presence of an initialized CEX2C. In the scenario described above, this CVT check failed. For our residency, we obtained an APARFIXTEST, which bypassed this CVT check. We expect there will be a resolving APAR created shortly after we conclude the residency.



Data Encryption tool installation and customization

In this chapter we discuss the installation procedure of Data Encryption for IMS and DB2 Databases Tool.

We first introduce the major techniques for encryption key generation and key management, prerequisite to the tool usage, by using the Integrated Cryptographic Service Facility (ICSF).

We then describe how a DBA would go about defining the encryption EDITPROC for a DB2 table.

This chapter contains the following sections:

- ▶ Generation of an encryption EDITPROC
- ▶ DB2 encryption implementation scenario for the DBA

13.1 Generation of an encryption EDITPROC

Integrated Cryptographic Service Facility (ICSF) is a component of z/OS, and is designed to transparently use the available cryptographic functions, whether CPACF or Crypto Express2, to balance the workload and help address the bandwidth requirements of your applications.

ICSF supports the Advanced Encryption Standard (AES) algorithm for data privacy. This updated algorithm provides stronger encryption. Key lengths of 128 bits, 192 bits and 256 bits are supported, depending on the class of System z processor used. If running on a z9, 128-bit AES encryption is available, the z10 customer can also use 192- and 256-bit AES clear key encryption. Secure key AES is available if running on IBM System z10 Enterprise Class and IBM System z10 Business Class processors and IBM System z 9 Enterprise Class and IBM System z9 Business Class processors with the appropriate microcode.

In the following example, we illustrate the way to generate a clear key for use by the Data Encryption for IMS and DB2 Databases Tool.

The cryptographic hardware (also known as the coprocessor) available to your applications depends on your processor or server model. z/OS ICSF supports the Crypto Express2 Feature. This feature is available on IBM System z9 Enterprise Class, IBM System z9 Business Class, IBM System z10 Enterprise Class, and IBM System z10 Business Class. It can be configured as a coprocessor or as an accelerator.

By using ICSF, you can generate clear keys or secure keys using either the Key Generator Utility Program (KGUP) or the key generation APIs. KGUP stores the key it generates in the CKDS. In our example, we illustrate how to define the CKDS (cryptographic key data set).

With ICSF callable services, you can generate a variety of cryptographic keys for use on your system or distribution to other systems. You can also develop key distribution protocols by using both secret key and public key cryptographic methods.

- ▶ Secret key distribution system:

You must first share a secret key with the system to which you intend to distribute keys. This is a major drawback with secret key distribution systems.

- ▶ With public key cryptography:

You encrypt the keys you are distributing under the receiver's public key. The receiver decrypts the keys by using the receiving system's private key. Public key encryption provides methods for key distribution and authentication.

As exploited by the Data Encryption for IMS and DB2 Databases Tool, we will only be using the symmetric key implementation of ICSF, used on a local level, so there is no requirement for key pairs or distribution of public or private keys.

13.1.1 Generate a Clear Key using ICSF

Figure 13-1 on page 301 shows the main panel for ICSF. In the following examples we create a CLRDES key using the ICSF ISPF interface. Data Encryption Standard (DES) is a method for encrypting information. CLRDES key is a clear DES key.

DES data-encrypting keys, also known as data keys, can be single-length, double-length, or triple-length. Data Encryption Algorithm is known as the DEA, the DES algorithm, or DES. We use the term DES as our reference to this algorithm during our illustrations.

All DES and AES keys, except the master keys, can be stored in the CKDS. We define the CKDS as part of our example. There are several methods you can use to enter keys into the CKDS.

```
CSF@PRIM ----- Integrated Cryptographic Service Facility-----
OPTION ==> 8
Enter the number of the desired option.

 1 COPROCESSOR MGMT - Management of Cryptographic Coprocessors
 2 MASTER KEY MGMT - Master key set or change, CKDS/PKDS Processing
 3 OPSTAT           - Installation options
 4 ADMINCNTL       - Administrative Control Functions
 5 UTILITY          - ICSF Utilities
 6 PPINIT          - Pass Phrase Master Key/CKDS Initialization
 7 TKE             - TKE Master and Operational Key processing
 8 KGUP           - Key Generator Utility processes
 9 UDX MGMT        - Management of User Defined Extensions

Licensed Materials - Property of IBM
5694-A01 Copyright IBM Corp. 1989, 2008. All rights reserved.
US Government Users Restricted Rights - Use, duplication or
disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Press ENTER to go to the selected option.
Press END   to exit to the previous menu.
```

Figure 13-1 Option 8 - Key Generator Utility processes

The key generator utility program (KGUP) is used to generate and maintain keys in the cryptographic key data set (CKDS). The CKDS stores DATA keys, MAC keys, PIN keys, and transport keys.

To execute KGUP, ICSF must be active, master keys must be loaded on the cryptographic coprocessors (which is now optional), the user must have the relevant access, and the CKDS must be successfully initialized.

The KGUP feature can be used to perform the following tasks:

- ▶ Generate or enter new keys
- ▶ Maintain current CKDS entries by either deleting or renaming those entries

When you enter a clear key value, KGUP does not encipher the clear key. Rather, it places the clear key into the CKDS. Because entering clear keys may endanger security, ICSF must be in special secure mode before you can enter a clear key by using KGUP. Special secure mode allows you to use KGUP to enter clear keys.

Option 1 allows you to create or edit key tokens through the ADD, UPDATE, DELETE, RENAME, or SET KGUP control statement. See Figure 13-2.

```
CSFSAM00 ----- ICSF - Key Administration -----  
OPTION ==> 1  
  
Enter the number of the desired option.  
  
1 Create          - Create key generator control statements  
2 Dataset         - Specify datasets for processing  
3 Submit          - Invoke Key Generator Utility Program (KGUP)  
4 Refresh         - Activate an existing cryptographic key dataset  
  
Press ENTER to go to the selected option  
Press END  to exit to the previous panel
```

Figure 13-2 Key Utility Generator creation panel

Specify the name of the control statement input data set into which KGUP is to place the statements. The data set does not have to pre-exist. It is allocated dynamically by the ICSF application. If the data set does not exist, you are asked for information about a data set allocation panel. See Figure 13-3.

```
CSFSAE10 ---- ICSF - KGUP Control Statement Dataset Specification -----  
COMMAND ==>  
  
Enter control statement input dataset (DDNAME = CSFIN)  
  
Dataset Name ==> 'paolr5.KGUP.control' _____  
Volume Serial ==> _____ (if uncatalogued)  
  
Press ENTER to open or create and open specified dataset  
Press END  to exit to the previous panel
```

Figure 13-3 Specifying the KGUP data set name

You can also use a partitioned data set. If you use a PDS, specify a member name as addition.

If you specify NOPREFIX in your TSO profile, specify the fully qualified data set name within apostrophes.

See Figure 13-4.

```
CSFSAE11 ----- ICSF - Allocation -----  
COMMAND ==>  
  
DATASET NAME: PAOLOR5.KGUP.CONTROL  
Dataset cannot be found. Specify allocation parameters below.  
  
VOLUME SERIAL    ==> _____ (Blank for authorized default volume) *  
GENERIC UNIT     ==> _____ (Generic group name or unit address) *  
SPACE UNITS      ==> BLOCK_____ (BLKS, TRKS, or CYLS)  
PRIMARY QUANTITY ==> 5_____ (In above units)  
SECONDARY QUANTITY ==> 0_____ (In above units)  
DIRECTORY BLOCKS ==> 0_____ (Zero for sequential data set)  
RECORD FORMAT    ==> FB  
RECORD LENGTH    ==> 80  
BLOCK SIZE       ==> 3200__ (In multiples of record length)  
EXPIRATION DATE  ==> _____ (Format is YYDDD)  
  
( * Only one of these fields may be specified)  
  
Press ENTER to allocate specified dataset and continue  
Press END to exit to the previous panel without allocating
```

Figure 13-4 Example of input fields to specify KGUP data set name and attributes

Once you enter the relevant information in order for ICSF to create the data set dynamically, it is empty. The next example shows how to update the KGUP data set.

Note: You only see the next panel after the KGUP control data set is successfully created. This is not an option from the main ICSF menu.

The control data set name that you previously specified (or as in our example, created) will be displayed. In the Data Set Name field, the control statement input data set that you specified previously is displayed. You can also change the name of the data set. ICSF appends any new control statements you create to the data set specified in the field.

Option 1 (Maintain) allows you to create an ADD, UPDATE, or DELETE control statement

See Figure 13-5.

```
CSFCSM00 ----- ICSF - KGUP Control Statement Menu ----- OPENED - EMPTY
OPTION ==> 1

Storage dataset for control statements (DDNAME = CSFIN)

Dataset Name: PAOLOR5.KGUP.CONTROL

Enter the number of the desired option above.

1 Maintain      - Create ADD, UPDATE, or DELETE control statements
2 Rename       - Create statement to RENAME entry label
3 Set          - Create a statement to SET installation data
4 Edit         - Edit the statement storage dataset

Press ENTER to go to the selected option
Press END   to exit to the previous panel
```

Figure 13-5 Updating the KGUP data set

Once the data set has been allocated you can proceed. When you choose the Maintain, Rename, or Set option, you access the panels to create the control statement you need. When you create a control statement, the statement is placed in the specified control statement input data set. To edit the control statements that are stored in this data set, you need to choose the Edit option.

On the panel, you need to fill out the input fields to create the ADD, UPDATE, or DELETE control statement that you want KGUP to process. Each field on the panel corresponds to a control statement keyword. The panel helps you to create a complete, syntactically correct ADD, UPDATE, or DELETE control statement.

See Figure 13-6.

```
CSFCSE10 ----- ICSF - Create ADD, UPDATE, or DELETE Key Statement -----
COMMAND ==>
Specify control statement information below

Function ==> _____ ADD, UPDATE, or DELETE
Algorithm ==> DES DES or AES
Key Type ==> _____ Outtype ==> _____ (Optional)
Label ==> _____
Group Labels ==> NO_ NO or YES
or Range:
Start ==> _____
End ==> _____

Transport Key Label(s)
==> _____
==> _____
or Clear Key ==> NO_ NO or YES
Control Vector ==> YES NO or YES
Length of Key ==> ___ For DES: 8, 16 or 24 For AES: 16, 24, or 32
Key Values ==>
_____, _____, _____, _____
Comment Line ==> _____
Press ENTER to create and store control statement
Press END to exit to the previous panel without saving
```

Figure 13-6 Creating and storing the key control statements

Specify the action that you want KGUP to perform in the Function field. Specify ADD to create a new CKDS entry, UPDATE to change a CKDS entry, or DELETE to erase a CKDS entry. Create ADD and UPDATE control statements to either import or generate a key. When KGUP generates a key, KGUP also generates the complement of the key.

Specify the type of key record that you would like KGUP to process in the Key Type field. See Figure 13-7.

```
----- Help for Create ADD, UPDATE, or DELETE Key Statement -----  
COMMAND ==>>
```

In the Key Type field, specify the type of key record that you want KGUP to process. You can specify the following:

- CLRAES (for clear AES encipher/decipher key)
- **CLRDES (for clear encipher/decipher key)**
- DATA (for encipher/decipher key)
- DATAM (for double length MAC generation key)
- DATAMV (for double length MAC verification key)
- DATAXLAT (for ciphertext translate key)
- EXPORTER (for exporter key encrypting key)
- IMPORTER (for importer key encrypting key)
- IPINENC (for input PIN encrypting key)
- MAC (for MAC generation key)
- MACVER (for MAC verification key)
- NULL (for NULL key record)
- OPINENC (for output PIN encrypting key)
- PINGEN (for PIN generation key)
- PINVER (for PIN verification key)

Leave the field blank to access the Key Type Selection panel.

Figure 13-7 Help for the KEY field - all the key options

In our example we use the CLRDES key. See Figure 13-8.

```
CSFCSE10 ----- ICSF - Create ADD, UPDATE, or DELETE Key Statement -----
COMMAND ==>
Specify control statement information below

Function ==> ADD__      ADD, UPDATE, or DELETE
Algorithm ==> DES      DES or AES
Key Type ==> CLRDES__  Outtype ==> _____ (Optional)
Label ==> SG24.7720.00.CLEAR.KEY.01_____
Group Labels ==> NO_   NO or YES
or Range:
Start ==> _____
End ==> _____

Transport Key Label(s)
==> _____
==> _____
or Clear Key ==> NO_   NO or YES
Control Vector ==> YES NO or YES
Length of Key ==> 24_ For DES: 8, 16 or 24 For AES: 16, 24, or 32
Key Values ==>
_____, _____, _____, _____
Comment Line ==> _____
Press ENTER to create and store control statement
Press END to exit to the previous panel without saving
```

Figure 13-8 Creating the KGUP Key statement

Specify the keylabel name that you want KGUP to process in the Label field. To specify one label, enter the label in the Label field and enter NO in the Group Labels field. More than one label can be specified by entering YES in the Group Labels field.

In the Length of Key field, specify the length of the key you want to generate. If you want to generate a single length key for a double-length key type, specify 8 or 16 respectively. In our scenario, we want to use a triple DES key, so the correct length is 24 bytes.

See Figure 13-9.

```
CSFCSE10 ----- ICSF - Create ADD, UPDATE, or DELETE K          SUCCESSFUL UPDATE
COMMAND ==>
Specify control statement information below

Function ==> ADD__      ADD, UPDATE, or DELETE
Algorithm ==> DES      DES or AES
Key Type ==> CLRDES__  Outtype ==> _____ (Optional)
Label ==> _____
Group Labels ==> NO_   NO or YES
or Range:
Start ==> _____
End ==> _____

Transport Key Label(s)
==> _____
==> _____
or Clear Key ==> NO_   NO or YES
Control Vector ==> YES NO or YES
Length of Key ==> 24_ For DES: 8, 16 or 24 For AES: 16, 24, or 32
Key Values ==>
_____, _____, _____, _____
Comment Line ==> _____
Press ENTER to create and store control statement
Press END to exit to the previous panel without saving
```

Figure 13-9 Successful Update of the key.

In Figure 13-10 we browse the control file to view the contents

```
Menu Utilities Compilers Help
-----
BROWSE PAOLOR5.KGUP.CONTROL1 Line 00000000 Col 001 080
Command ==> Scroll ==> CSR
***** Top of Data *****
ADD TYPE(CLRDES) LENGTH(24),
LAB(SG24.7720.00.CLEARDES.KEY.03)
***** Bottom of Data *****
```

Figure 13-10 Contents of the Control file - showing the key label

Figure 13-11 shows a print listing of the contents of the updated control file.

```

Process  Options  Help
-----
View          PAOLOR5.KGUP.CONTROL                      Top of 3
Command ==>                                     Scroll PAGE
          Col 1      Insert Length 80          Record AT TOP      Format CHAR
          -----10-----2-----3-----4-----5-----6-----7--
***** **** Top of data ****
000001 ADD TYPE(CLRDES) ,
000002 LAB(SG24.7720.00.CLEAR.KEY.01)
***** **** End of data ****

```

Figure 13-11 Contents of the updated KGUP Control file using IBM File Manager for z/OS

We use option 2 - Dataset to specify the input and output process data sets. This is the data set that KGUP uses for processing. See Figure 13-12.

```

CSFSAM00 ----- ICSF - Key Administration -----
OPTION ==> 2

Enter the number of the desired option.

1 Create          - Create key generator control statements
2 Dataset         - Specify datasets for processing
3 Submit         - Invoke Key Generator Utility Program (KGUP)
4 Refresh        - Activate an existing cryptographic key dataset

Press ENTER to go to the selected option
Press END  to exit to the previous panel

```

Figure 13-12 Creating other needed data sets for the KGUP

An initialized, CKDS must exist. None of the other data sets have to exist. In our example, both the CSFCKDS “PAOLOR5,SC63.CSFCKDS” and the control data set. “PAOLOR5.KGUP.CONTROL” already exists and it was not needed to add them, because they were added dynamically.

The Diagnostics Data Set Name is a data set to which KGUP is to write a copy of each control statement and any diagnostic information. It is not mandatory to allocate this data set.

If you specified a control statement input data set on the KGUP Control Statement Data Set Specification panel, the data set name appears in the Control Statement Input Data Set Name field on this panel. If you change the data set name on this panel, it automatically changes on the KGUP Control Statement Data Set Specification panel.

The Key Output Data Set Name field is a data set which is to contain the complementary key information that KGUP writes when it generates a key. The Control Statement Output Data Set Name contains the complementary key control statements that KGUP writes when it generates a key.

See Figure 13-13.

```
CSFSAE20 ----- ICSF - Specify KGUP Datasets -----  
COMMAND ==>  
  
Enter dataset names for all cryptographic files.  
Cryptographic Keys      (DDNAME = CSFCKDS)  
  Dataset Name ==> 'PAOLOR5.SC63.CSFCKDS' _____  
  
Control Statement Input  (DDNAME = CSFIN)  
  Dataset Name ==> 'PAOLOR5.KGUP.CONTROL' _____  
  Volume Serial ==> _____ (if uncataloged)  
  
Diagnostics              (DDNAME = CSFDIAG) (use * for printer)  
  Dataset Name ==> 'paolor5.kgup.diag' _____  
  Volume Serial ==> _____ (if uncataloged)  
  
Key Output               (DDNAME = CSFKEYS)  
  Dataset Name ==> 'paolor5.kgup.csfkeys' _____  
  Volume Serial ==> _____ (if uncataloged)  
  
Control Statement Output (DDNAME = CSFSTMNT)  
  Dataset Name ==> 'paolor5.kgup.csfstmnt' _____  
  Volume Serial ==> _____ (if uncataloged)  
  
Press ENTER to set the dataset names. Press END to exit to the previous panel.
```

Figure 13-13 Specify the names of the data sets needed for KGUP processing

Once you hit ENTER, the data set names will be set. See Figure 13-14.

```
...  
//KGUP EXEC PGM=CSFKGUP,PARM=('SSM')  
//CSFCKDS DD DSN=PAOLOR5.SC63.CSFCKDS,  
// DISP=OLD  
//CSFIN DD DSN=PAOLOR5.KGUP.CONTROL,  
// DISP=OLD  
//CSFDIAG DD DSN=PAOLOR5.KGUP.DIAG,  
// DISP=(,CATLG,CATLG),UNIT=SYSDA,  
// DCB=(RECFM=FBA,LRECL=133,BLKSIZE=13300),  
// SPACE=(TRK,(220,10),RLSE)  
//CSFKEYS DD DSN=PAOLOR5.KGUP.CSFKEYS,  
// DISP=(,CATLG,CATLG),UNIT=SYSDA,  
// DCB=(RECFM=FB,LRECL=208,BLKSIZE=3328),  
// SPACE=(TRK,(60,10),RLSE)  
//CSFSTMNT DD DSN=PAOLOR5.KGUP.CSFSTMNT,  
// DISP=(,CATLG,CATLG),UNIT=SYSDA,  
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),  
// SPACE=(TRK,(60,10),RLSE)  
...
```

Figure 13-14 JCL generated by ICSF for KGUP

You can now submit this job and check the output for success.

13.2 DB2 encryption implementation scenario for the DBA

We use the IBM Data Encryption for IMS and DB2 Databases Tool to implement a successful encryption of a DB2 table.

In the previous chapter we created the cryptographic keys in the ICSF data set. This is a requirement for the Data Encryption for IMS and DB2 Databases Tool. This data set is the key repository where both the Clear Keys and Secure Keys are stored. To initialize and use the CKDS, ICSF may require that a Secure Key device is available. Even if Clear Keys is used only to protect the databases, a Secure Key device may be required to initialize and use the CKDS. The Secure Key devices also require that a master key be loaded before the Secure Key functions are available.

The IBM Data Encryption for IMS and DB2 Databases Tool uses data encryption keys that are stored inside the CKDS (a special RACF protected data set) and may be extracted, encrypted, and decrypted through the use of the CEX2C hardware feature. This is an optional feature. This is the most secure key management solution available on the market today, and is the only solution that is EAL5 and FIPS 140 Level 4 compliant (these are security standards maintained by the NIST and NSA).

13.2.1 Creating the DB2 user exit routine by using ISPF panels

Figure 13-15 shows the main menu for the IBM Data Encryption for IMS and DB2 Databases Tool.

```
DATA ENCRYPTION FOR IMS AND DB2 DATABASES - PK75337

Command ==> 1

Select an OPTION to continue or END to exit

OPTION . .

      1      Build a standalone encryption DB2 EDITPROC or IMS exit

      2      Build a DB2 compression/encryption EDITPROC

      3      Build an IMS compression/encryption exit
```

Figure 13-15 Main menu for Data Encryption for IMS and DB2 Databases Tool

Use option 1 to enter a key label identifier that the IBM exits use. This is also known as the encryption key label. Our key labels were defined using the Integrated Cryptographic Service Facility (ICSF) Key Generation Update (KGUP) utility in the previous section. The key labels have a maximum length of 64 characters.

See Figure 13-16.

```

                                DATA ENCRYPTION FOR IMS AND DB2 DATABAS      Enter required field

Command ==>
Press ENTER to continue or END to exit
Specify ICSF encryption key to be implemented.
Key label . . SG24.7720.00.CLEAR.KEY.02

Specify 1 for IMS or DB2 SECURE KEY
          2 for DB2 CLEAR KEY
          3 for IMS DES CLEAR KEY
          4 for IMS AES CLEAR KEY. . . 2
DBMS . . . . . DB2 (IMS or DB2)

Specify encryption JCL parameters.
Jobcard . . //PAOLOR5L JOB (999,POK),DET,
          . . //          NOTIFY=&SYSUID,
          . . //          CLASS=A,
          . . //          MSGCLASS=X,
          . . //          MSGLEVEL=(1,1),REGION=0M
CSF lib . . CSF.SCSFMODE
ZAP lib . . SYS1.LINKLIB
SMP lib . . DEC.V1R1MO.SDECLMDO
Exit lib . . DB9A9.SDSNEXIT
Exit name . . SG7720X2

```

Figure 13-16 Data Encryption for IMS and DB2 Databases Tool Main Menu

Enter a DB2 CLEAR KEY exit (DECENA00) by selecting option 2. The encryption key labels are usually defined by the security analyst who installs or administers ICSF. The key can either be defined as a DATA (for Secure Key) or CLRDES (for Clear Key) key type, with a key length of 8, 16, or 24 bytes (based on the database risk requirements).

For the CSF (ICSF) library, enter the name of the library that contains the ICSF modules (CSNBENC, CSNBDEC, CSNBSYE, CSNBSYD, and CSNBKRR). The ZAP library name is the library that contains AMASPZAP (load module zap) program. The tool uses the zap program to enter the encryption key label into the exit that you build.

For the SMP library, enter the library name that contains the IBM encryption routines. This is normally the same library where you installed the encryption product. The Exit library is the library that contains the DB2 EDITPROC exit, once it is built.

The Exit name is the name of your (user specified) encryption exit. The name can be 8 bytes long.

See Figure 13-17.

```
...
//LINK      EXEC PGM=IEWL,PARM='LIST,XREF,RENT'
//SYSPRINT  DD SYSOUT=*
//SYSUDUMP  DD SYSOUT=*
//SDECLMDO  DD DSN=DEC.V1R1M0.SDECLMDO,DISP=SHR
//SCSFMODE  DD DSN=CSF.SCSFMODO,DISP=SHR
//SYSUT1    DD UNIT=SYSDA,SPACE=(1024,(50,50))
//SYSLMOD   DD DSN=DB9A9.SDSNEXIT(SG7720X1),DISP=SHR
//SYSLIN    DD *
            ENTRY DECENAO0
            INCLUDE SDECLMDO(DECENAO0)
            INCLUDE SCSFMODO(CSNBKRR)
            NAME SG7720X1(R)
//*
//*
//BATHTSO   EXEC PGM=IKJEFT01,DYNAMNBR=25,REGION=0M,COND=EVEN
//SYSLIB    DD DISP=SHR,DSN=DB9A9.SDSNEXIT
//ISPLLIB   DD DISP=SHR,DSN=SYS1.LINKLIB
//ISPPLIB   DD DISP=SHR,DSN=ISP.SISPPENU
//ISPSLIB   DD DISP=SHR,DSN=ISP.SISPSENU
//ISPMLIB   DD DISP=SHR,DSN=ISP.SISPMENU
//ISPTLIB   DD DISP=SHR,DSN=ISP.SISPTENU
//SYSPROC   DD DISP=SHR,DSN=ISP.SISPCLIB
//SYSEXEC   DD DISP=SHR,DSN=ISP.SISPEXEC
//          DD DISP=SHR,DSN=DEC.V1R1M0.SDECCEXE
//ISPTABL   DD DISP=SHR,DSN=PAOLOR5.SC63.ICSF.ISPPROF
//ISPPROF   DD DISP=SHR,DSN=PAOLOR5.SC63.ICSF.ISPPROF
//SYSTSPRT  DD SYSOUT=*
//ISPLOG    DD SYSOUT=*,DCB=(BLKSIZE=800,LRECL=80,RECFM=FB)
//SYSTSIN   DD *
            PROFILE PREFIX(PAOLOR5)
            ISPSTART CMD(%DECENC02 DB2 SG7720X1 -
            SG24.7720.00.CLEAR.KEY.01
            )
/*
/*
/*
/* yyyyyyyyyy = encryption key to be used, e.g., ICSFDB2KEY
/*          Can be up to 64 characters maximum length
```

Figure 13-17 Generated JCL which builds the encryption exit.

The generated JCL invokes TSO in batch to link and builds the encryption exit routine.

In our example, in Figure 13-17, the name of our exit (as supplied in the ISPF panels) is shown (SG7720X1) and the Clear Key name (SG24.7720.00.CLEAR.KEY.01).

13.2.2 Implementing DB2 encryption

Once you have build the DB2 encryption exit routine, as described in the previous section, you are now ready to perform the DBA DB2 tasks to encrypt the DB2 table rows. Based on the presence of this EDITPROC on the table, DB2 determines that the EDITPROC exit is required. DB2 now loads the exit. Next, DB2 calls the exit and passes it the un-encrypted row. The exit invokes ICSF services, passing the user-defined data encryption key label (provided by the exit, and in our example it is called "SG24.7720.00.CLEAR.KEY.01") and the un-encrypted row.

Our key label refers to a CLRDES key type and which we predefined by using ICSF. Data encryption can now be implemented by performing a database unload and reload activities. Once the data is unloaded, update the DDL and specify the EDITPROC name. You now drop and re-create your table with the DDL containing the EDITPROC. After you unload a table, and prior to reloading it, specify the EDITPROC option and the name of your customized user exit routine to encrypt a table that has been specified for encryption.

13.2.3 Max record size

As shown in Table 13-1, the maximum record size for each page size depends on the size of the table space and whether you specified the EDITPROC clause. The EDITPROC uses 10 bytes for its definition.

Table 13-1 Maximum record size

EDITPROC	4-KB page	8-KB page	16-KB page	32-KB page
NO	4056	8138	16330	32714
YES	4046	8128	16320	32704



Data encryption scenarios

In this chapter we discuss some typical encryption scenarios using the Data Encryption for IMS and DB2 Databases Tool.

The three major choices when using the Data Encryption tool are as follows:

- ▶ Master key protected CKDS
- ▶ Clear-key-only Cryptographic Key Data Set (HCR7751)
- ▶ Compression and encryption

14.1 Master key protected CKDS

Whenever there are requirements to support ICSF-based encryption key management of keys other than TDES or AES clear keys, there is the hardware requirement for one or more CEX2C features to be available and with valid AES or TDES Master keys loaded. When using clear keys, the Crypto Express2 is not used to encrypt/decrypt data. From a Data Encryption Tool perspective, the Crypto Express2 is only used for initializing the CKDS when there are secure keys (for other purposes) stored in the CKDS.

14.1.1 Clear key

As discussed earlier, TDES encryption, while still viewed as secure by the NIST, is rapidly being replaced by AES encryption, where the supporting hardware is available to exploit the most favorable performance characteristics. For many customers, including those who have a requirement to share encrypted data outside of the glass data center, TDES is still considered the industry standard, and in an environment where a customer wishes to establish a standard common denominator for encryption algorithms across different organizations, TDES is still the logical choice.

For our scenario, we first review the process of building the TDES clear key using the ICSF key generation utility program KGUP. The generation of key materials is typically the responsibility of designated security analysts, also known as key officers. The choice of the key type, key length, and the actual key label value are all made by the key officer in the execution of the KGUP utility as shown in Figure 14-1.

```
. ----- ICSF - Create ADD, UPDATE, or DELETE Key Statement ----- .
. COMMAND ==> .
. Specify control statement information below .
. .
.   Function ==> add_      ADD, UPDATE, or DELETE .
.   Algorithm ==> DES      DES or AES .
.   Key Type ==> clrdes_   Outtype ==> _____ (Optional) .
.   Label ==> CLEAR.KEY.FOR.SECURE.EXIT _____ .
.   Group Labels ==> NO_   NO or YES .
. or Range: .
.   Start ==> _____ .
.   End ==> _____ .
. .
.   Transport Key Label(s) .
.   ==> _____ .
.   ==> _____ .
. or Clear Key ==> NO_     NO or YES .
.   Control Vector ==> YES NO or YES .
.   Length of Key ==> 24_ For DES: 8, 16 or 24 For AES: 16, 24, or 32 .
.   Key Values ==> .
.   _____ , _____ , _____ , _____ .
.   Comment Line ==> _____ .
. Press ENTER to create and store control statement .
. Press END to exit to the previous panel without saving .
. .
. .
```

Figure 14-1 KGUP clear key specification

Once the key has been generated, the EDITPROC needs to be prepared through a JCL procedure. This can be created by using either the ISPF dialog box provided by the Data Encryption for IMS and DB2 Databases Tool, or by using a modified version of sample JCL that can be located in the Data Encryption for IMS and DB2 Databases Tool samplib, named 'HLQ.SDECSAMP'. For a DB2 clear key encryption EDITPROC, the samplib member that should be modified is named DECDB2CK. Figure 14-2 shows an example of how we modified our samplib member.

Note that the keylabel value specified in the BATCHTSO step is identical to the keylabel specified in the KGUP panel. This specification is case sensitive, and positional, so make sure that the key label, including trailing blanks, is exactly 64 bytes in length and uppercase because the KGUP utility will fold it to uppercase.

```

//LINK      EXEC PGM=IEWL,PARM='LIST,XREF,RENT'           00003500
//SYSPRINT  DD SYSOUT=*                                  00003600
//SYSUDUMP  DD SYSOUT=*                                  00003700
//SDECLMDO  DD DSN=DEC.V1R1MO.SDECLMDO,DISP=SHR         00003800
//SCSFMODE  DD DSN=CSF.SCSFMODO,DISP=SHR               00003900
//SYSUT1    DD UNIT=SYSDA,SPACE=(1024,(50,50))         00004000
//SYSLMOD   DD DSN=DB9A9.SDSNEXIT(CLEAREXT),DISP=SHR   00004100
//SYSLIN    DD *                                        00004200
            ENTRY DECENA00                               00004300
            INCLUDE SDECLMDO(DECENA00)                  00004400
            INCLUDE SCSFMODO(CSNBKRR)
            NAME CLEAREXT(R)                             00004700
//*                                                 00004800
//*                                                 00004900
//BATCHTSO  EXEC PGM=IKJEFT01,DYNAMNBR=25,REGION=0M,COND=EVEN 00005000
//SYSLIB    DD DISP=SHR,DSN=DB9A9.SDSNEXIT             00005100
//ISPLLIB   DD DISP=SHR,DSN=SYS1.LINKLIB               00005200
//ISPLLIB   DD DISP=SHR,DSN=ISP.SISPPENU              00005300
//ISPLIB    DD DISP=SHR,DSN=ISP.SISPSENU              00005400
//ISPLIB    DD DISP=SHR,DSN=ISP.SISPMENU              00005500
//ISPTLIB   DD DISP=SHR,DSN=ISP.SISPTENU              00005600
//SYSPROC   DD DISP=SHR,DSN=ISP.SISPCLIB              00005700
//SYSEXEC   DD DISP=SHR,DSN=ISP.SISPEXEC              00005800
//          DD DISP=SHR,DSN=DEC.V1R1MO.SDECCEXE
//ISPTABL   DD DISP=SHR,DSN=PAOLR5.SC63.ICSF.ISPPROF   00005900
//ISPPROF   DD DISP=SHR,DSN=PAOLR5.SC63.ICSF.ISPPROF   00006000
//SYSTSPRT  DD SYSOUT=*                                00006100
//ISPLOG    DD SYSOUT=*,DCB=(BLKSIZE=800,LRECL=80,RECFM=FB) 00006200
//SYSTSIN   DD *                                        00006300
            PROFILE PREFIX(PAOLR5)
            ISPSTART CMD(%DECENC02 DB2 CLEAREXT -
            CLEAR.KEY.FOR.ENCRYPT.TABLE )

```

Figure 14-2 Modified DECDB2CK from SDECSAMP

Because this keylabel is associated with a clear TDES key, we used the SDECSAMP member DECDB2CK as our example. This is important to, as the associated link-edit control statements will include the DECENA00, which is the Data Encryption for IMS and DB2 Databases Tool clear key encryption routine. Care must be taken to not use the wrong member in SDECSAMP, as there are other members to prepare different forms of EDITPROCs. See Table 14-1 on page 318.

Table 14-1 SDECSAMP JCL members

Member Name	Description
DECDB2CK	DB2 Clear Key EDITPROC
DECDB2JB	Undocumented Samplib Member
DECDB2DV	DB2 Compression Editproc and Driver
DECDB2SK	DB2 Secure Key EDITPROC
DECIMSCB	IMS Clear Key AES Exit
DECIMSK	IMS Clear Key TDES Exit
DECIMSDV	IMS Compression Editproc and Driver
DECIMSSK	IMS Secure Key Exit
DECIMSJB	Undocumented Samplib Member

Once prepared, the EDITPROC will be link-edited into the DB2 APF authorized library. For most installations of DB2, this will be the DB2HLQ.SDSNEXIT library. Once linked into the APF library, the first SQL statement that invokes the associated table will cause DB2 to MVS load the EDITPROC into private storage. Once this occurs, any subsequent change and re-link of the EDITPROC will not take effect until the next re-cycle of the DB2 subsystem.

Attention: Most installations provide a RACF access level (UACC) of read to the SDSNEXIT load library. Be aware that the keylabel values are visible in the link-edited EDITPROC module. For this reason, if this is viewed as a security exposure, one suggestion is to create a separate APF authorized library, and use this library solely for Data Encryption for IMS and DB2 Databases Tool generated EDITPROCS. This data set could then be assigned the RACF default access UACC(NONE), and then only allow DB2 UACC (READ) and designated key officers UACC(WRITE). However, if third-party tools or utilities that read the underlying Linear VSAM data sets for table space access, those might also need access to the EDITPROC APF library.

EDITPROC definition and review

Edit routines are assigned to a table by the EDITPROC clause of CREATE TABLE. An edit routine receives the entire row of the base table in DB2 format. It can transform that row when it is stored by an INSERT or UPDATE SQL statement, or by the LOAD utility. It also receives the transformed row during retrieval operations and must change it back to its original form.

Restriction: You cannot use an edit routine on a table that contains a LOB or a ROWID column.

The transformation the edit routine performs on a row is called *edit-encoding*. The same routine is used to undo the transformation when rows are retrieved, called *edit-decoding*.

The edit routine can encode the entire row of the table, including any index keys. However, index keys are extracted from the row before the encoding is done. Therefore, index keys are stored in the index in edit-decoded form. That means, for a table with an edit routine, index keys in the table are edit-coded. Index keys in the index are not edit-coded.

Important: Changes made to the row do not affect or change value stored in the index, so although the underlying base table is encrypted, any associated indexes are not.

To name an edit routine for a table, use the EDITPROC clause of the CREATE TABLE statement, followed by the name of the routine. If you plan to use an edit routine, specify it when you create the table. In operation, the routine is loaded on demand. You cannot add an edit routine to a table that already exists. You must drop the table and re-create it. Also, you cannot alter a table with an edit routine to add a column. Again, you must drop the table and re-create it, and presumably also alter the edit routine in some way to account for the new column.

Important: Once an EDITPROC is implemented, any further ALTER to that table has to be performed as part of a DROP and re-create. Tables being considered for encryption that are volatile from a schema change perspective might not be good candidates for encryption because of this restriction.

An edit routine is invoked to edit-code a row whenever DB2 inserts or updates one, including inserts made by the LOAD utility. It is invoked after any date routine, time routine, or field procedure. If there is also a validation routine, the edit routine is invoked after the validation routine. Any changes made to the row by the edit routine do not change entries made in an index. The same edit routine is invoked to edit-decode a row whenever DB2 retrieves one. On retrieval, it is invoked before any date routine, time routine, or field procedure. If retrieved rows are sorted, the edit routine is invoked before the sort. An edit routine is not invoked for a DELETE operation without a WHERE clause that deletes an entire table in a segmented table space.

Editproc and encryption implementation

Once the EDITPROC is prepared, the next step is the implementation and subsequent first time encryption process. Because there is a DB2-enforced restriction on the use of ALTER to add the EDITPROC declaration through schema evolution, this will require a *destructive alteration* to implement the EDITPROC. This approach is supported by many of the popular database administration products. In our environment we used the DB2 Administration tool to build the script necessary to support this alteration. Other third-party administration products provide a similar capability. In the absence of such a tool, this can be performed manually. In general the process is as follows:

1. Ensure that a good and consistent full image copy has been created prior to starting the implementation. Remember, that at some point in the process, we will DROP the table, so plan for a recovery back to this point in the event fallback is dictated.
2. Capture ALL the DDL associated with the Table, as upon the DROP being executed, all of the associated catalog elements will also be lost. The list of associated elements can include:
 - Indexes
 - Foreign Key relationships
 - Views
 - Synonyms
 - Triggers
 - Authorizations granted to tables and views
3. Depending on your DSNZPARM autobind parameters, you may also want to collect the packages and plans which reference the table so that rebind can be done after the drop and subsequent recreate
4. Run UNLOAD to create the SYSREC unloaded row file. This will be subsequently used to LOAD the table, once the create is performed.
5. DROP the table.

6. Using the DDL collected above, add the EDITPROC declarative to the table DDL and then execute the DDL to rebuild the table. At this point, the Data Encryption for IMS and DB2 Databases Tool generated EDITPROC is not related to the table.
7. Execute the LOAD utility, this will on each ISRT invoke the EDITPROC and the row will be converted from unencrypted text to ciphertext.
8. Take a second full image copy of the encrypted table space. This is now your new recovery point. At this point you are running in an encrypted environment for this table.

14.1.2 Encryption from a data management perspective

As described above, the EDITPROC will be invoked on Insert prior to the row being externalized by DB2, and on Select directly prior to the row being presented to the application in an SQL result set. Figure 14-3 describes the process flow in detail.

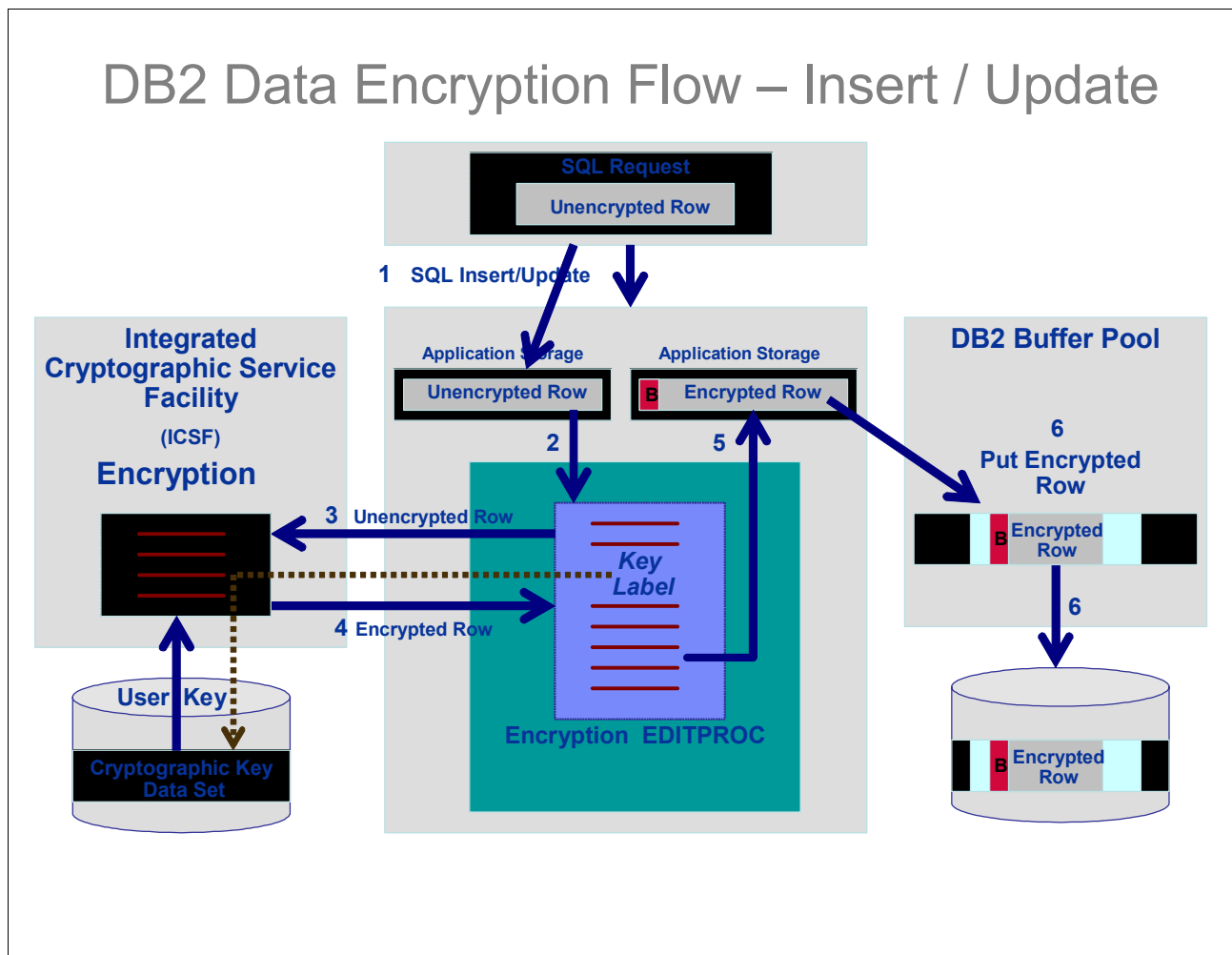


Figure 14-3 Insert processing with encrypting EDITPROC

1. SQL is passed to DB2 from an SQL request, in this example either an ISRT or UPDT request. At this point, the row image is stored in the application buffer in clear-text format.
2. DB2 invokes the named EDITPROC, passing a parameter list which points to the unencrypted row in application storage.
3. At this point, the process is different for clear keys and secure keys, as follows:

- For clear key, the EDITPROC passes the keylabel to ICSF through the use of the CSNBKRR routine, this uses the key record read by callable service to copy an internal key token from the in-storage CKDS to application storage. If the key is located, then a normal return code will be returned to the EDITPROC routine. If there is a keylabel mismatch, or if there are other ICSF environmental issues, ICSF will return one of several return codes, and the EDITPROC will return SQLCODE -652. Then the EDITPROC will invoke the CPACF instruction KMC passing the key token retrieved by CSNBKRR and the unencrypted row image. The row is transformed by the appropriate encryption algorithm (AES or TDES).
 - For secure key, the process is a little different, on an encryption request, the EDITPROC will call the CSNBENC encoding (encrypting) routine of ICSF. This routine will then perform the secure key encryption process inside the confines of the CEX2C cryptographic feature.
4. Once encryption is performed, return is give to the EDITPROC, and the row image stored in application storage has now been transformed into ciphertext.
 5. The application commits.
 6. Once the row is updated, the row image as stored on the DB2 buffer pool page is also now encrypted in ciphertext. Ultimately, the updated data page is externalized to disk, and at that point, the data at rest row image is also encoded in ciphertext.

Encryption and the DB2 Recovery Log

One other interesting characteristic with the EDITPROC implementation of encryption is that when rows are transformed by an EDITPROC, it is the transformed row image that is written to the DB2 recovery log. This ensures that the archived recovery log, when recovery versions are shipped to an offsite recovery location, are themselves encrypted. To demonstrate this, we conducted the following scenario.

Tip: It is often necessary to document that data is actually transformed to ciphertext to demonstrate compliance with audit requirements. The procedure below can be used for this purpose.

Accessing an encrypted table, in this example, using a TDES clear key implementation, we first constructed the insert statement shown in Example 14-1.

Example 14-1 Sample INSERT statement

```
INSERT INTO PAOLOR5.GLWTEPA
      ( EMP_NO,PROJ_NO,ACT_NO,EMPTIME,EMSTDATE,EMENDATE,CREATED_TS,CREATED_BY,UPDATED_TS,UPDATED_BY )
VALUES( 9930, 1, 2, 500, '2009-01-02', '2009-01-03', '2009-01-19-17.19.53.12353 3', 'PAOXXX',
'2009-01-19-17.19.53.123533', 'PAOYYYY')
```

This table will be used further in this discussion. Using the IBM DB2 Administration Tool, we will show the different columns declared in the table definition, shown in Figure 14-4 on page 322. We will reference these columns later in this discussion.

```

. DB2 Admin -- DB9A Columns in Table PAOLOR5.NEWTEPA ----- Row 1 of 10
. 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
. * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
. -----
. EMP_NO 1 INTEGER 4 0 N N N -1
. PROJ_NO 2 INTEGER 4 0 N N N -1
. ACT_NO 3 INTEGER 4 0 N N N -1
. EMPTIME 4 DECIMAL 5 2 Y Y N -1
. EMSTDATE 5 DATE 4 0 N N N -1
. EMENDATE 6 DATE 4 0 Y Y N -1
. CREATED_TS 7 TIMESTMP 10 0 N Y N -1
. CREATED_BY 8 CHAR 8 0 N Y N -1
. UPDATED_TS 9 TIMESTMP 10 0 N Y N -1
. UPDATED_BY 10 CHAR 8 0 N Y N -1
. ***** END OF DB2 DATA *****

```

Figure 14-4 Table definition used in the log scenario

We determined the RBA range we needed to search the DB2 recovery log for an eligible logical unit of work (LUOW). We issued a display log command. In our environment this is **-DB9A DISPLAY LOG**.

This resulted in the display shown in Figure 14-5.

```

DSNJ370I -DB9A DSNJC00A LOG DISPLAY
CURRENT COPY1 LOG = DB9AU.LOGCOPY1.DS03 IS 85% FULL
CURRENT COPY2 LOG = DB9AU.LOGCOPY2.DS03 IS 85% FULL
      H/W RBA = 00178FA426BD
      H/O RBA = 00178DDCFFFF
      FULL LOGS TO OFFLOAD = 0 OF 6
      OFFLOAD TASK IS (AVAILABLE)
DSNJ371I -DB9A DB2 RESTARTED 15:28:39 FEB 21, 2009
      RESTART RBA 001776C7C000
      CHECKPOINT FREQUENCY 500000 LOGRECORDS
      LAST SYSTEM CHECKPOINT TAKEN 18:08:56 FEB 21, 2009
DSN9022I -DB9A DSNJC001 '-DIS LOG' NORMAL COMPLETION

```

Figure 14-5 Display of DB9A Log RBA status

Using this information, we coded the appropriate DSN1LOGP control statements. To gather this information, we query the DB2 catalog to get the object identifiers. The SQL statement shown in Example 14-2 can be used to locate the object identifiers needed by DSN1LOGP.

Example 14-2 Catalog query to extract DBID, PSID, and OBID

```

SELECT DBID, PSID FROM SYSIBM.SYSTABLESPACE
WHERE NAME= 'GLWSEPA' AND DBNAME= 'PAOLOR5';
SELECT NAME, OBID FROM SYSIBM.SYSTABLES WHERE TSNAME= 'GLWSEPA'
AND CREATOR= 'PAOLOR5' ;

```

With the RBA log range and the object identifiers, we built JCL and control statements for a DSN1LOGP summary report. In our sample database and table space, the DBID was 368 and the OBID was 28. The control statements and JCL are shown in Figure 14-6.

```
//STEP1 EXEC PGM=DSN1LOGP
//STEPLIB DD DSN=DB9A9.SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSSUMRY DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//*SDS DD DSN=DB9AU.BSDS01,DISP=SHR
//ARCHIVE DD DSN=DB9AU.ARCHLOG1.A0000340,DISP=SHR
//SYSIN DD *
RBASTART (00178DDDD0000) RBAEND(00178FA426BD)
DBID (368) OBID(28) SUMMARY(YES)
```

Figure 14-6 DSN1LOGP Summary report JCL and Control Statements

Upon running this, we then were able to locate the unit of recovery (UR) that was related to our insert statement previously executed. We executed this SQL statement using the IBM DB2 Administration Tool, which is associated with plan ADB, and with an authorization ID of PAOLOR5. Figure 14-7 shows the output from this request.

```
DSN1151I DSN1LPRT UR CONNID=TSO CORRID=PAOLOR5 AUTHID=PAOLOR5
PLAN=ADB
START DATE=09.052 TIME=18:23:58 DISP=COMMITTED
INFO=COMPLETE
STARTRBA=00178FA3B5C6 ENDRBA=00178FA3C207 STARTLRSN=C3C865A7CD55
ENDLRSN=C3C865A80210
NID=* LUWID=USIBMSC.SCPDB9A.C3C8644DDA94.0001
COORDINATOR=* PARTICIPANTS=*
DATA MODIFIED:
DATABASE=0170=PAOLOR5 PAGE SET=001C=GLWSEPA
DATABASE=0170=PAOLOR5 PAGE SET=001F=GLWXEPA1
DATABASE=0170=PAOLOR5 PAGE SET=0021=GLWXEPA2
DATABASE=0170=PAOLOR5 PAGE SET=0023=GLWXEPA3
```

Figure 14-7 Output from DSN1LOGP summary report

In looking at this report, we can use the plan, authid, and corrid fields to verify this is the UR in which we are interested. Also, note that this UR updated the table in table space GLWSEPA, and also resulted in three additional index updates. To run the detail report, what we are interested in is the LUWID, highlighted in Figure 14-7.

Using this identifier, we code a second set of control statements for DSN1LOGP. These are shown in Figure 14-8 on page 324. Notice that we have specified the LUWID that was highlighted in Figure 14-7.

```

//STEP1 EXEC PGM=DSN1LOGP
//STEPLIB DD DSN=DB9A9.SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSSUMRY DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//*SDS DD DSN=DB9AU.BSDS01,DISP=SHR
//ARCHIVE DD DSN=DB9AU.ARCHLOG1.A0000340,DISP=SHR
//SYSIN DD *
LUWID(USIBMSC.SCPDB9A.C3C8644DDA94.0001)
SUMMARY(NO)

```

Figure 14-8 DSN1LOGP with LUWID for detail reporting

This gives us the resulting output. For clarity, we have only included the UNDO/REDO record from the INSERT statement. This is shown in Figure 14-9.

```

00178FA3B656 TYPE( UNDO REDO ) URID(00178FA3B5C6)
          LRSN(C3C865A7CD55) DBID(0170) OBID(001C) PAGE(00000002)          18:23:58 09.052
          SUBTYPE(INSERT IN A DATA PAGE) CLR(NO) PROCNAME(DSNISGRT)

*LRH* 00830090 06000001 0E800017 8FA3B5C6 00178FA3 B5C60626 00178FA3 B5C6C3C8 * c          t F t F t FCH
          65A7CD55 0000 * x
*LG** 88017000 1C000000 02000017 85D098F1 4C00 *h          e q1<
0000 004B5039 001D0008 00004300 1D393B31 F3403259 C4741FF9 6FFD1827 F329FD39 * .&          3 D 9? 3
0020 4F9C742B DC8BE130 06B48D6D 838BB66C A259A50A 5A7C3038 7A47B342 2740C4DE * |          _c %s v !@ : D
0040 F588A4B5 A43B72E5 76E01E *5hu u V

```

Figure 14-9 DSN1LOGP UNDO/REDO of an encrypted table

Compare the above results with an INSERT operation on an unencrypted version of the table, described previously in Figure 14-4 on page 322 with the same schema. Notice the values of the columns CREATED_BY and UPDATED_BY, shown in Figure 14-10.

```

00178FE6954C TYPE( UNDO REDO ) URID(00178FE694BC)
          LRSN(C3C992014CCD) DBID(0170) OBID(001C) PAGE(00000002)          16:47:43 09.053
          SUBTYPE(INSERT IN A DATA PAGE) CLR(NO) PROCNAME(DSNISGRT)

*LRH* 00830090 06000001 0E800017 8FE694BC 00178FE6 94BC0626 00178FE6 94BCC3C9 * c          Wm Wm Wm CI
          92014CCD 0000 *k <
*LG** 88017000 62000000 02000017 8FE5FED3 4C00 *h          V L<
0000 004B503A 00660000 00004300 663A8000 115C8000 00018000 000200F5 00002009 * .&          * 5
0020 01020020 09010320 09011917 19531235 33D7C1D6 E7E7E7E7 40200901 19171953 *          PAOXXXX
0040 123533D7 C1D6E8E8 E8E840 * PAOYYYY

```

Figure 14-10 DSN1LOGP UNDO/REDO of a cleartext table

The values in the cleartext log record are clearly visible, while the values in the encrypted log record are undecipherable. The key point to this exercise is that we have demonstrated that the contents of the recovery log are encrypted. As a result, when these assets leave the data center to the recovery site, if for some reason the logcopy data sets are lost or stolen, no data is compromised. The corresponding log records are encrypted and protected from unauthorized use.

Encrypted recovery log and log processing

While protecting the contents of the recovery log by encrypting the log records associated with updates to encrypted tables, there is some potential for problems occurring with certain types of log processing.

Because of the fact that log records have been encoded using the EDITPROC and its associated key value, for log records to be processed in a meaningful way, the same processing needs to be performed when reading encrypted log records. When using the IFCID 306 log read, DB2 will ensure that for log records associated with tables with EDITPROC declared, the EDITPROC will be driven before the log record is passed back through the log read.

Some possible areas where this might be of concern would be with tools that perform following tasks:

- ▶ Tools such as the IBM Log Analysis Tool which read the DB2 recovery log for reporting purposes.
- ▶ Data replication or propagation products such as IBM WebSphere Information Integration Q Replication.
- ▶ Recovery products such as IBM Recovery Expert.

All such products from IBM have been written to exploit IFCID 306 and as such are completely compatible with the Data Encryption for IMS and DB2 Databases Tool. For other third-party solutions, verify with your vendor their ability to operate on encrypted log records.

DSN1COPY and data movement

One other consideration with the use of encryption concerns the use of DSN1COPY as a mechanism to move data from one table space to another, either within the same DB2 or with a separate DB2 subsystem as target. Care must be taken, particularly with the foreign DB2 scenario, to ensure that the same encryption environment is in place. Failure to ensure this will result in inaccessible data. To demonstrate this, the following scenario has been constructed. Using DSN1COPY, we moved encrypted data from one table space into a second table space, ensuring that the same encryption EDITPROC was available, then verified that we were able to successfully access the data.

The first table was constructed with DDL shown in Figure 14-11.

```
CREATE TABLE PAOLOR5.GLWTEPA
  (EMP_NO           INTEGER NOT NULL,
   PROJ_NO         INTEGER NOT NULL,
   ACT_NO          INTEGER NOT NULL,
   EMPTIME         DECIMAL(5, 2) WITH DEFAULT NULL,
   EMSTDATE        DATE NOT NULL,
   EMENDATE        DATE WITH DEFAULT NULL,
   CREATED_TS      TIMESTAMP NOT NULL WITH DEFAULT,
   CREATED_BY      CHAR(8) FOR SBCS DATA NOT NULL
     WITH DEFAULT,
   UPDATED_TS      TIMESTAMP NOT NULL WITH DEFAULT,
   UPDATED_BY      CHAR(8) FOR SBCS DATA NOT NULL
     WITH DEFAULT,
   CONSTRAINT GLWPEPA
     PRIMARY KEY (PROJ_NO,
                  ACT_NO,
                  EMP_NO))
IN PAOLOR5.GLWSEPA
EDITPROC CLEARTEXT
AUDIT ALL
```

Figure 14-11 DDL for encrypted table - original

In a separate table space we created another table, identical to the original table, and with the same EDITPROC. The DDL is shown in Figure 14-12.

```
CREATE TABLE PAOLOR5.NEWTEPA
  (EMP_NO           INTEGER NOT NULL,
   PROJ_NO         INTEGER NOT NULL,
   ACT_NO          INTEGER NOT NULL,
   EMPTIME         DECIMAL(5, 2) WITH DEFAULT NULL,
   EMSTDATE        DATE NOT NULL,
   EMENDATE        DATE WITH DEFAULT NULL,
   CREATED_TS      TIMESTAMP NOT NULL WITH DEFAULT,
   CREATED_BY      CHAR(8) FOR SBCS DATA NOT NULL
     WITH DEFAULT,
   UPDATED_TS      TIMESTAMP NOT NULL WITH DEFAULT,
   UPDATED_BY      CHAR(8) FOR SBCS DATA NOT NULL
     WITH DEFAULT,
   CONSTRAINT GLWPEPA
     PRIMARY KEY (PROJ_NO,
                  ACT_NO,
                  EMP_NO))
IN PAOLOR5.NEWSEPA
EDITPROC CLEARTEXT
AUDIT ALL
```

Figure 14-12 DDL for encrypted table - clone

In comparing the two sets of DDL, the only differences are the table names and the table space names, those being unique to allow for uniqueness within the same DB2.

Next, we determined the DBID, PSID, and OBID identifiers that are necessary to run the DSN1COPY utility with the OBIDXLAT parameter specified. We used the SQL shown in Figure 14-13 to determine the proper identifiers.

```

---- SOURCE ENCRYPTED TABLESPACE ----
SELECT DBID, PSID FROM SYSIBM.SYSTABLESPACE
WHERE NAME= 'GLWSEPA' AND DBNAME= 'PAOLOR5';
SELECT NAME, OBID FROM SYSIBM.SYSTABLES WHERE TSNAME= 'GLWSEPA'
AND CREATOR= 'PAOLOR5' ;
---- TARGET ENCRYPTED TABLESPACE ----
SELECT DBID, PSID FROM SYSIBM.SYSTABLESPACE
WHERE NAME= 'NEWSEPA' AND DBNAME= 'PAOLOR5';
SELECT NAME, OBID FROM SYSIBM.SYSTABLES WHERE TSNAME= 'NEWSEPA'
AND CREATOR= 'PAOLOR5' ;

```

Figure 14-13 SQL to determine object identifiers

Running this SQL, we determine the following identifiers associated with the source and target objects, Table 14-2 shows the results of this SQL query.

Table 14-2 Object Identifiers

	DBID	PSID	OBID
Source	368	28	29
Target	368	98	102

The DBID identifiers are the same value. We created the target table space in the same database as the source table space, for illustrative purposes.

We ran the DSN1COPY utility to copy and translate the pages from source to target, using the JCL in Figure 14-14.

```

//EXECUTE EXEC PGM=DSN1COPY,PARM='OBIDXLAT,RESET'
//STEPLIB DD DSN=DB9A9.SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=DB9AU.DSNDBC.PAOLOR5.GLWSEPA.I0001.A001,
// DISP=SHR
//SYSUT2 DD DSN=DB9AU.DSNDBC.PAOLOR5.NEWSEPA.I0001.A001,
// DISP=SHR
//SYSXLAT DD *
368,368
28,98
29,102
/*

```

Figure 14-14 DSN1COPY JCL example

Once the DSN1COPY completed, we ran a small SQL SELECT to verify that we were able to retrieve and decrypt the data successfully.

Using DSNTEP2, we selected representative sampling of the data shown in Figure 14-15.

```
SELECT * FROM PAOLR5.NEWTEPA FETCH FIRST 10 ROWS ONLY
```

	EMP_NO	PROJ_NO	ACT_NO	EMPTIME	EMSTDATE	EMENDATE	CREATED_TS
1_	5003	1	1	0.50	2002-11-12	2002-12-04	2005-12-19-17.19.53.123533
2_	5004	1	1	0.50	2002-11-12	2002-12-04	2005-12-19-17.19.53.123533
3_	5003	1	2	0.50	2002-12-05	2002-12-27	2005-12-19-17.19.53.123533
4_	5004	1	2	0.50	2002-12-05	2002-12-27	2005-12-19-17.19.53.123533
5_	5003	1	3	0.50	2002-12-28	2003-01-19	2005-12-19-17.19.53.123533
6_	5004	1	3	0.50	2002-12-28	2003-01-19	2005-12-19-17.19.53.123533
7_	5003	1	4	0.50	2003-01-20	2003-02-11	2005-12-19-17.19.53.123533
8_	5004	1	4	0.50	2003-01-20	2003-02-11	2005-12-19-17.19.53.123533
9_	5003	1	5	0.50	2003-02-12	2003-03-06	2005-12-19-17.19.53.123533
10_	5004	1	5	0.50	2003-02-12	2003-03-06	2005-12-19-17.19.53.123533

Figure 14-15 DSNTEP2 output from SELECT

14.1.3 Encryption confirmation techniques

When implementing encryption into a production environment, in many cases it will be necessary to document the effects of encryption to ensure that the data is indeed actually being encrypted. This section discusses situations where you can use DSN1PRNT to demonstrate the effects of encryption on tablespace data and image copy data sets. We discussed a technique for using DSN1LOGP for recovery log validation previously in “Encrypted recovery log and log processing” on page 325

Using DSN1PRNT to confirm encryption results

One other area where the audit or security personnel might ask for documentation is a demonstration that encryption has actually occurred and the tables are protected by encryption. The best technique for demonstrating this is to use DSN1PRNT and print one or more pages from the linear VSAM data set. For our purposes, we are going to show a typical data page print formatted with DSN1PRNT in an unencrypted environment. In Figure 14-16 we are asking for the first five data pages to be printed.

```
//PAOLR5M JOB (999,P0K),CSF-CKDS,
//      NOTIFY=&SYSUID,
//      CLASS=A,
//      MSGCLASS=X,
//      MSGLEVEL=(1,1),
//      REGION=0M
//RUNPRNT EXEC PGM=DSN1PRNT,PARM='PRINT(1,5),FORMAT,EXPAND'
//STEPLIB DD DSN=DB9A9.SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=DB9AU.DSNDBC.A130X997.LIPS.I0001.A001,DISP=SHR
```

Figure 14-16 DSN1PRNT JCL to print data pages of VSAM LDS

After running the DSN1PRNT job against the unencrypted table, we can see that data exists in unencrypted text format, as seen in Figure 14-17.

```

PAGE: # 00000002 -----
DATA PAGE: PGCOMB='00'X  PGLOGRBA='00179BA6B0C4'X  PGNUM='00000002'X  PGFLAGS='00'X  PGFREE=1572
           PGFREE='0624'X  PGFREEP=2518  PGFREEP='09D6'X  PGHOLE1='0000'X  PGMAXID='02'X  PGNANCH=0
PGTAIL: PGIDFREE='00'X  PGEND='E'
ID-MAP FOLLOWS:
01 0014 04F9

RECORD: XOFFSET='0014'X  PGSFLAGS='00'X  PGSLTH=1253  PGSLTH='04E5'X  PGSOBD='0003'X  PGSBID='01'X
00033930 3000A30 30383030 30353631 35000630 30303031 30000354 414E0008 ..900..0080005615..000010..TAN..
5343484C 45494552 00063136 30323137 00083230 30303033 32300008 53443030 SCHLEIER..160217..20000320..SD00
30303030 00085344 30303030 30300009 42454E43 484D4152 4B000430 30303100 0000..SD000000..BENCHMARK..0001.
04303030 31000120 00012000 01200001 20F00000 00001000 00025354 00025354 .0001.. .. .. ..ST..ST
F00001F0 0001F000 00000001 0000F000 00000001 00000002 4B47F000 00000001 ..
00000001 4C000120 000120F0 00F00000 01200001 20000832 30303131 323230F0 ....L.. .. .. ..20011220.
00000000 1000001C 4D617465 7269616C 206E6963 68742064 6973706F 72656C65 .....Material nicht disporele
76616E74 00012000 01200006 30303030 30300001 20000120 000A3130 30303030 vant.. ..000000.. ..100000
30303032 00063030 30303130 00015800 06303030 30303000 01410004 30303031 0002..000010..X..000000..A..0001
00043030 30310001 20000120 00012000 01200001 20000120 00033630 31000330 ..0001.. .. .. ..601..0
30300001 20000120 00012000 04484157 41000120 00012000 01200001 20000120 00.. .. ..HAWA.. .. ..

```

Figure 14-17 DSN1PRNT unencrypted text example

Next, we recreated the table with an encrypting EDITPROC specified, and reloaded the data that converted the table rows from unencrypted text into cipher text. Running DSN1PRNT a second time shows that the data has been converted into ciphertext. This is demonstrated in Figure 14-18.

```

PAGE: # 00000002 -----
DATA PAGE: PGCOMB='00'X  PGLOGRBA='000000000000'X  PGNUM='00000002'X  PGFLAGS='00'X  PGFREE=1516
           PGFREE='05EC'X  PGFREEP=2574  PGFREEP='0A0E'X  PGHOLE1='0000'X  PGMAXID='02'X  PGNANCH=0
PGTAIL: PGIDFREE='00'X  PGEND='E'
ID-MAP FOLLOWS:
01 0014 04F9

RECORD: XOFFSET='0014'X  PGSFLAGS='00'X  PGSLTH=1253  PGSLTH='04E5'X  PGSOBD='0003'X  PGSBID='01'X
5C32C913 E8FFD983 FE16EB76 615DB56F 45CA1EB5 1BEFB72D 18C8BDB6 E239DC66 \2.....va}.oE.....-.....9.f
7E6EE4CE 4F4DDED8 80E755D6 9B162680 DC2D8C54 BD3286CB 1BDF5C01 C26CB894 .n..OM....U...&..-.T.2....\..1..
FDF1C0D3 CF128EF5 E4A68821 FEC43845 FF4CA503 C4D2AFAD 24909239 762ADC69 .....!..8E.L.....$.9v*.i
C4D60D7E EF88CE7C 5D1D8A11 13AC9DD5 367D3E52 9FB970F1 B06D9CF2 DD39343F .....}......6.>R..p..m...94?
1C237B2E 77F015E9 8854FBC7 539FC0A0 33C294D2 FB46CF40 5D185FCB 0DF60CD1 #..w....T..S...3....F.@}_.....
D698EC67 24A31B20 38FF9104 9CF6E6CB ED03D71E F34FF233 3A8D8AB0 B645D384 ...g$. 8.....0.3:....E..
EE3B0F3F 8E9983E8 0649D998 65AE57E4 5C076758 73B1C741 63EDB1AE EDA217E6 ;.?....I..e.W.\.gXs..Ac.....
5CBB4554 C91B19CA F4045EEF 061E2DA4 C4A521B2 BB75D1AA A3D5BCFB 61EB2B05 \.ET.....-.....!..u.....a.+
1E53CBE8 OCD3FD26 5855E538 1905BCDE D49F81E8 A5FBE948 4618706F 65F06394 .S.....&XU.8.....HF.poe.c.
4DOC6093 7FE4A1A8 776F1D10 D54AE3AB OC43B056 B43071DE CCA4911B 9F7AAC55 M.....wo...J...C.V.0q.....z.U
E755CD4C 0CFCE5DC FD89F0FB 5FCFA808 C6406A6D 62ABA870 510CAF23 23438B95 .U.L....._.....@jmb..pQ..##C..

```

Figure 14-18 DSN1PRNT encrypted data page example

Verifying image copy encryption

With encrypted tables, standard recovery assets including the DB2 recovery log and DB2 image copy data sets are also encrypted. The recovery log record is encrypted by virtue of the log image reflecting the row after the EDITPROC is driven. The DB2 image copy data set is encrypted due to the nature of the image copy utility being a page level operation. Again, using DSN1PRNT, it is easy to document this for purposes of satisfying audit and security requirements.

Using the standard IBM COPY utility, we made a full shrlevel reference image copy of our encrypted table space. Next, using the JCL shown in Figure 14-19, we print the first five pages of our image copy data set.

```
//PAOLOR5M JOB (999,P0K),CSF-CKDS,
//          NOTIFY=&SYSUID,
//          CLASS=A,
//          MSGCLASS=X,
//          MSGLEVEL=(1,1),
//          REGION=OM
//RUNPRNT EXEC PGM=DSN1PRNT,PARM='PRINT(1,5),FORMAT'
//STEPLIB DD DSN=DB9A9.SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=PAOLOR5.DB9A.IC002.PAOLOR5.LIPS,DISP=SHR
```

Figure 14-19 DSN1PRNT using image copy input

As shown in the output in Figure 14-20, the image copy pages are also protected through encryption.

```
PAGE: # 00000002 -----
DATA PAGE: PGCOMB='00'X PGLGRBA='000000000000'X PGNUM='00000002'X PGFLAGS='00'X PGFREE=1516
           PGFREE='05EC'X PGFREEP=2574 PGFREEP='0A0E'X PGHOLE1='0000'X PGMAXID='02'X PGNANCH=0
PGTAIL: PGIDFREE='00'X PGEND='E'
ID-MAP FOLLOWS:
01 0014 04F9

RECORD: XOFFSET='0014'X PGSFLAGS='00'X PGSLTH=1253 PGSLTH='04E5'X PGSOBD='0003'X PGSBID='01'X
5C32C913 E8FFD983 FE16EB76 615DB56F 45CA1EB5 1BEFB72D 18C8BDB6 E239DC66 \2.....va}.oE.....-.....9.f
7E6EE4CE 4F4DDED8 80E755D6 9B162680 DC2D8C54 BD3286CB 1BDF5C01 C26CB894 .n..OM....U...&..-.T.2....\..1..
FDF1CD03 CF128EF5 E4A68821 FEC43845 FF4CA503 C4D2AFAD 24909239 762ADC69 .....!.8E.L.....$.9v*.i
C4D60D7E EF88CE7C 5D1D8A11 13AC9DD5 367D3E52 9FB970F1 B06D9CF2 DD39343F .....}......6.>R..p..m...94?
1C237B2E 77F015E9 8854FBC7 539FC0A0 33C294D2 FB46CF40 5D185FCB 0DF60CD1 .#.w....T..S...3....F.@}_.....
D698EC67 24A31B20 38FF9104 9CF6E6CB ED03D71E F34FF233 3A8D8AB0 B645D384 ...g$. 8.....0.3:....E..
EE3B0F3F 8E9983E8 0649D99B 65AE57E4 5C076758 73B1C741 63EDB1AE EDA217E6 .;?...I..e.W.\.gXs..Ac.....
5CBB4554 C91B19CA F4045EEF 061E2DA4 C4A521B2 BB75D1AA A3D5BCFB 61EB2B05 \.ET.....-...!..u.....a.+
1E53CBE8 0CD3FD26 5855E538 1905BCDE D49F81E8 A5FBE948 4618706F 65F06394 .S.....&XU.8.....HF.poe.c.
4DOC6093 7FE4A1A8 776F1D10 D54AE3AB 0C43B056 B43071DE CCA4911B 9F7AAC55 M.....wo...J...C.V.0q.....z.U
E755CD4C 0CFEC5DC FD89F0FB 5FCFA808 C6406A6D 62ABA870 510CAF23 23438B95 .U.L....._.....@jmb..pQ..##C..
```

Figure 14-20 DSN1PRNT encrypted image copy page

14.1.4 Secure key

Similar to our explorations with clear key encryption, we executed a similar set of workloads running in a secure key environment. There was an expectation that due to the use of the CEX2C crypto feature, to which access is performed across an I/O bus, there was a significant increase in elapsed time.

While the preparation of a secure key EDITPROC is similar to that performed for clear key, there are some important differences.

Secure key specification

The first major distinction is in the form of a key that is prepared by the security or key officer using ICSF and the KGUP utility. We see in Figure 14-21 on page 331 that we are using a key type of DATA. Earlier forms of TDES clear keys were generated with a key type of CLRDES.

Data indicates to ICSF that this is a secure key, and will be wrapped or encrypted under the master key by the CEX2C feature before being stored in the CKDS.

```

----- ICSF - Create ADD, UPDATE, or DELETE K          SUCCESSFUL UPDATE
COMMAND ==>
Specify control statement information below

Function ==> ADD_      ADD, UPDATE, or DELETE
Algorithm ==> DES      DES or AES
Key Type ==> DATA_    Outtype ==> _____ (Optional)
Label ==> SG24.7720.00.SECUREDES.KEY.01_____
Group Labels ==> NO_   NO or YES
or Range:
Start ==> _____
End ==> _____

Transport Key Label(s)
==> _____
==> _____
or Clear Key ==> NO_    NO or YES
Control Vector ==> YES NO or YES
Length of Key ==> 16_ For DES: 8, 16 or 24 For AES: 16, 24, or 32
Key Values ==>
_____, _____, _____, _____
Comment Line ==> _____
Press ENTER to create and store control statement
Press END to exit to the previous panel without saving

```

Figure 14-21 Secure key KGUP example

The second difference when preparing a secure key EDITPROC is in the JCL used to generate it. The correct JCL sample to use for secure key processing is member DECDB2SK located in the *HLQ.SDECSAMP* data set. Notice that this JCL member includes a different set of ICSF and Data Encryption for IMS and DB2 Databases Tool modules as part of the link edit. Figure 14-22 on page 332 shows a modified sample that was used for our secure key scenario testing.

When looking at the secure key example, notice the link-edit include statements for CSNBENC and CSNBDEC. These are the ICSF API-based routines to perform secure key encryption (CSNBENC) and decryption (CSNBDEC). This is different from the clear key preparation where there were no API routines linked, because in a clear key environment, our generated EDITPROCs will use the KMC instruction supported on the hardware with CPACF.

Tip: When executing in a mixed key environment (clear and secure), the most common source of problems is using the incorrect JCL to prepare the EDITPROC. Ensure that when expecting a secure key exit, the above ICSF routines are linked into the EDITPROC and the key label and key type must match the EDITPROC. We describe this scenario and associated symptoms in 15.5.3, “Out-of-synch key labels” on page 360.

When the key token is created in the CKDS, the index is the key label concatenated to the key type. If you create a CLRDES key with a label of KEYLAB1, you must use the appropriate JCL to create a clear DES key.

```

//LINK      EXEC PGM=IEWL,PARM='LIST,XREF,RENT'          00003500
//SYSPRINT  DD SYSOUT=*                                  00003600
//SYSUDUMP  DD SYSOUT=*                                  00003700
//SDECLMDO  DD DSN=DEC.V1R1M0.SDECLMDO,DISP=SHR        00003800
//SCSFMODO  DD DSN=CSF.SCSFMODO,DISP=SHR               00003900
//SYSUT1    DD UNIT=SYSDA,SPACE=(1024,(50,50))         00004000
//SYSLMOD   DD DSN=DB9A9.SDSNEXIT(SG77SEC1),DISP=SHR   00004100
//SYSLIN    DD *                                        00004200
           ENTRY DECENCOO
           INCLUDE SDECLMDO(DECENCOO)
           INCLUDE SCSFMODO(CSNBENC)
           INCLUDE SCSFMODO(CSNBDEC)
           NAME SG77SEC1(R)                               00004700
/*                                                00004800
/*                                                00004900
//BATHTSO   EXEC PGM=IKJEFT01,DYNAMNBR=25,REGION=OM,COND=EVEN 00005000
//SYSLIB    DD DISP=SHR,DSN=DB9A9.SDSNEXIT              00005100
//ISPLLIB   DD DISP=SHR,DSN=SYS1.LINKLIB                00005200
//ISPPLIB   DD DISP=SHR,DSN=ISP.SISPPENU              00005300
//ISPSLIB   DD DISP=SHR,DSN=ISP.SISPSENU              00005400
//ISPMLIB   DD DISP=SHR,DSN=ISP.SISPMENU              00005500
//ISPTLIB   DD DISP=SHR,DSN=ISP.SISPTENU              00005600
//SYSPROC   DD DISP=SHR,DSN=ISP.SISPCLIB              00005700
//SYSEXEC   DD DISP=SHR,DSN=ISP.SISPEXEC              00005800
//          DD DISP=SHR,DSN=DEC.V1R1M0.SDECCEXE
//ISPTABL   DD DISP=SHR,DSN=PAOLOR5.SC63.ICSF.ISPPROF  00005900
//ISPPROF   DD DISP=SHR,DSN=PAOLOR5.SC63.ICSF.ISPPROF 00006000
//SYSTSPRT  DD SYSOUT=*                                  00006100
//ISPLLOG   DD SYSOUT=*,DCB=(BLKSIZE=800,LRECL=80,RECFM=FB) 00006200
//SYSTSIN   DD *                                        00006300
           PROFILE PREFIX(PAOLOR5)
           ISPSTART CMD(%DECENC02 DB2 SG77SEC1 -
           SG24.7720.00.SECUREDS.KEY.01
/*

```

Figure 14-22 Secure key JCL sample

14.1.5 AES 128 clear key

Unlike secure key, the AES implementation is controlled by the form of key generated by the key officer, and the same process used to prepare the EDITPROC for the clear TDES form of EDITPROC is used for AES.

Figure 14-23 on page 333 shows the preparation of an AES clear key and associated key label using KGUP facilities of ICSF. Notice the key type specified for this example is CLRAES.

```

----- ICSF - Create ADD, UPDATE, or DELETE K          SUCCESSFUL UPDATE
COMMAND ==>
Specify control statement information below

Function ==> ADD__      ADD, UPDATE, or DELETE
Algorithm ==> AES       DES or AES
Key Type ==> CLRAES__   Outtype ==> _____ (Optional)
Label ==> SG24.7720.00.CLEARAES.KEY.02 _____
Group Labels ==> NO_   NO or YES
or Range:
Start ==> _____
End   ==> _____

Transport Key Label(s)
====> _____
====> _____
or Clear Key          ==> NO_      NO or YES
Control Vector ==> YES NO or YES
Length of Key ==> 16_   For DES: 8, 16 or 24   For AES: 16, 24, or 32
Key Values          ==>
_____, _____, _____, _____
Comment Line ==> _____
Press ENTER to create and store control statement
Press END   to exit to the previous panel without saving

```

Figure 14-23 KGUP CLRAES key specification

For this implementation, we chose the 16-byte key length. This results in the generation of an AES 128-bit key.

Attention: When running on a z9 processor, the only AES clear key implementation that is supported is 128-bit. For clear AES 256-bit support, you need to be running on a z10. The reason for this is that the tool implements clear key encryption through the use of CPACF, and AES 256 on z9 is only supported on the software, not through CPACF, and is not exploited by the tool.

We will discuss more about AES 128 in the performance section, but in general the scenarios using TDES and AES clear key were identical, outside of the KGUP preparation described above.

14.2 Clear-key-only Cryptographic Key Data Set (HCR7751)

If you choose to run with an unprotected CKDS, as described in 12.3.3, “HCR7751 and CKDS operations without CEX2C” on page 295, then clear keys can be created and stored in the CKDS without the use of CEX2C. Other than the removal of the DES master key requirement, and the associated use of CEX2C, there are no other considerations for using EDITPROCS that perform clear key TDES or AES processing.

Restriction: With this configuration, there is no support for secure key encryption as implemented in the IBM Data Encryption for IMS and DB2 Databases Tool.

14.3 Compression and encryption

In this section we do not discuss index compression.

Your compressed data and your encrypted data are treated by DB2 in a similar way. See Table 14-3.

Table 14-3 DB2 treatment of compressed and encrypted data

On disk	In the buffer pool	In the log	In the copies	To the application
Encrypted	Encrypted	Encrypted	Encrypted	Un-encrypted
Compressed	Compressed	Compressed	Compressed	Un-compressed

The cost of encryption and compression is paid at the row level every time the row is passed from the buffer pool to the application. The cost of decrypting and decompressing is paid at the row level every time the row is passed from the application to buffer pool. Such overhead is reflected in the accounting class 2 DB2 CPU.

Compression is not a substitute for encryption. If you want to enable both with DB2, you should consider compressing your data sets before encrypting the data. In some cases, the compression method does not actually shorten the data. In those cases, the data is left uncompressed and readable. If you encrypt and compress your data, you should compress it first. After you obtain the maximum compression, encrypt the result. When you retrieve your data, first decrypt the data. After the data is decrypted, decompress the result.

Historically, as implemented with the Data Encryption for IMS and DB2 Databases Tool, encryption and compression were viewed as incompatible. The reason for this is due to the secure nature of the data transformation performed by encryption algorithms such as TDES or AES.

To be secure, one of the characteristics of all encryption algorithms is to ensure that when dealing with character data, each byte of cleartext, especially when encountering repeating characters or trailing blanks, is encoded as a unique non-repeating byte of ciphertext.

Compression, on the other hand, works best when encountering strings of repeating character data. In general, the highest degree of compression benefit is seen when applied against character data.

So, when implementing encryption using the Data Encryption for IMS and DB2 Databases Tool, the order of encryption versus compression is important to understand. The sequence of events from a DB2 implementation of compression is that EDITPROCs are invoked first, to perform any necessary row transformation, including in our case encryption through ICSF. Once the EDITPROC transformation is performed, the row is then passed to the DB2 compression mechanism to compress the row prior to being externalized.

In this scenario, what is presented to the DB2 compression routines is not cleartext with repeating characters or trailing blanks, but rather ciphertext with non-repeating unique hexadecimal representations. When applying the compression algorithms against data with ciphertext characteristics, there is zero compression benefit achieved.

Due to this mechanism, when encrypting data with Data Encryption for IMS and DB2 Databases Tool, the recommendation has been to turn off compression at the owning tablespace level using the COMPRESS NO tablespace attribute. The reason for this

recommendation was an acknowledgement that no compression benefit would be achieved based on the ciphertext nature of the tablespace data, and the additional overhead associated with DB2 compression would result in zero benefit.

As a result, in some cases, customers not only had to absorb the additional overhead associated with the implementation of encryption, but also saw additional performance deterioration due to the loss of compression. Some of these lost benefits are as follows:

- ▶ Disk space savings
- ▶ Increased buffer pool hit ratios due to higher row residency on each data page
- ▶ Reduced I/O associated with getpage activity, again related to more rows on a given data page

14.3.1 Compression support in Data Encryption for IMS and DB2 Databases Tool

To provide customers an option where encryption and compression can coexist, with favorable performance benefits, changes were introduced to Data Encryption for IMS and DB2 Databases Tool with PTF UK41766 (APAR PK69786) and follow on UK44642 (APAR PK81724). This maintenance delivers a mechanism where compression can be performed by the EDITPROC before the ICSF-based encryption is applied. This not only restores all of the benefits of compressed data described above, but as each compressed row is presented to ICSF, the amount of data needing encryption is reduced, and a small incremental performance benefit to the encryption cost is achieved.

The general approach, described in detail below, involved changes to the DB2 offline utility DSN1COMP. These changes were delivered as co-requisite PTFs. For V8, the function is delivered with UK41354, and for the V9 environment, with PTF UK41355. These changes include documentation changes to the corresponding utilities reference guide and are documented in the cover letter of APAR PK69786.

Other recent maintenance includes PTF UK44045 (APAR PK80254) and UK41773 (APAR PK75337)

DSN1COMP changes for Data Encryption for IMS and DB2 Databases Tool

In a normal DB2 implementation of compression, the offline utility DSN1COMP is used to perform a sample-based analysis of pages in a DB2 table space being considered for compression. To learn how much space you can save by compressing your data, run the DSN1COMP utility on your data sets. Message DSN1940I of DSN1COMP reports an estimate of the percentage of kilobytes that would be saved by using data compression. DSN1COMP will build a temporary compression dictionary and apply this dictionary to the target table space to derive its compression percentage estimates. When finished with this analysis, the compression dictionary is discarded

With UK41354/UK41355 applied, a new option has been added to DSN1COMP, which provides a mechanism to save the compression dictionary built by DSN1COMP in an object deck format. This dictionary is used as input into the compression process delivered by UK41766.

DSN1COMP has been changed so that the compression dictionary that is created in DSN1COMP for determining the compression report can be externalized and stored in a data set for further processing. The object module with the CSECT-name specified in option

EXTNDICT is written to the data set identified with the DSN1DICT DD statement. It can be up to 64 KB. This externalized compression dictionary can be included into a program through the linkage editor.

A new option EXTNDICT has been added for DSN1COMP. EXTNDICT specifies the eight character name of the link editable object deck built from the DSN1COMP-created compression dictionary. The name has to follow the z/OS naming conventions. Only alphanumeric characters (in upper-case) are allowed to build-up the name. The first character has to be non-numeric. The new keyword can be placed anywhere in the parameter list

Specifying option EXTNDICT requires providing a DSN1DICT DD statement in the DSN1COMP Job. DSN1DICT defines the output data set to which the generated object module is written and stored for follow-on processing. This data set has to be sequential or a member of a partitioned data set with the record-format fixed and a record-length of 80.

In Example 14-3, we specify that DSN1COMP is to externalize the DSN1COMP-generated compression dictionary. This is achieved by specifying option EXTNDICT, which requires that a DSN1DICT DD statement is provided. DSN1DICT identifies the output data set to which the generated object module is written and stored for follow-on processing.

Example 14-3 DSN1COMP with EXTNDICT parameter

```
//BUILD EXEC PGM=DSN1COMP,  
// PARM='DSSIZE(4G),EXTNDICT(DICTTEP2),ROWLIMIT(99999) '  
//STEPLIB DD DSN=DB9A9.SDSNLOAD,DISP=SHR  
//SYSPRINT DD SYSOUT=*  
//SYSUT1 DD DSN=DB9AU.DSNDBC.PAOLOR5.GLWSEPA.I0001.A001,  
// DISP=SHR  
//DSN1DICT DD DSN=PAOLOR5.DSNDB04.TSDEPT.COMP.DICT,  
// DISP=(,CATLG,DELETE),UNIT=SYSDA,  
// SPACE=(TRK,(8,4)),  
// DCB=(LRECL=80,BLKSIZE=4000,RECFM=FB)
```

Figure 14-24 shows the expected output from the utility execution upon successful completion of the DSN1COPY. In our example, we achieve an estimate savings of approximately 30%.

```

DSN1999I START OF DSN1COMP FOR JOB PAOLOR5L BUILD
DSN1998I INPUT DSNAME = DB9AU.DSNDBC.PAOLOR5.GLWSEPA.I0001.A001      , VSAM
DSN1944I DSN1COMP INPUT PARAMETERS
          4,096  DICTIONARY SIZE USED
              0  FREEPAGE VALUE USED
              5  PCTFREE VALUE USED
          99,999 ROWLIMIT REQUESTED
              ESTIMATE BASED ON DB2 LOAD METHOD
              255 MAXROWS VALUE USED
          DICTTEPA  EXTERNAL DICTIONARY OBJECT NAME

DSN1940I DSN1COMP COMPRESSION REPORT
          6,543  KB WITHOUT COMPRESSION
          4,528  KB WITH COMPRESSION
              30  PERCENT OF THE BYTES WOULD BE SAVED

          2,991  ROWS SCANNED TO BUILD DICTIONARY
          99,999 ROWS SCANNED TO PROVIDE COMPRESSION ESTIMATE
          4,096  DICTIONARY ENTRIES

              69  BYTES FOR AVERAGE UNCOMPRESSED ROW LENGTH
              49  BYTES FOR AVERAGE COMPRESSED ROW LENGTH

              64  DICTIONARY PAGES REQUIRED
          1,786  PAGES REQUIRED WITHOUT COMPRESSION
          1,347  PAGES REQUIRED WITH COMPRESSION
              24  PERCENT OF THE DB2 DATA PAGES WOULD BE SAVED

DSN1937I DSN1COMP TXT-DECK DICTTEP2 BUILT          1,173  RECORDS WRITTEN
DSN1994I DSN1COMP COMPLETED SUCCESSFULLY,        1,788  PAGES PROCESSED

```

Figure 14-24 DSN1COMP output example

Implementing compression support in Data Encryption for IMS and DB2 Databases Tool

As shown above, the first step to implementing compression support with Data Encryption for IMS and DB2 Databases Tool is to build a compression dictionary. As described, the output from DSN1COMP is an object deck. In our example, it is a sequential data set.

The next step in the process is to prepare the compression dictionary and the different forms of exits necessary for compression support. In our example, we chose to create separate partition data sets for each component. This is to help show the process and help you understand the different components. In summary, the process consists of the following steps:

1. Using DSN1COMP create a object deck compression dictionary (described in “DSN1COMP changes for Data Encryption for IMS and DB2 Databases Tool” on page 335).
2. Link-edit the compression dictionary into load module format
3. Create and link-edit a compression standalone EDITPROC

4. Create and link-edit an encryption standalone EDITPROC
5. Create a driver EDITPROC which includes the compression and the encryption EDITPROC

Attention: It is the EDITPROC with the compressing and encrypting EDITPROC that is declared in the DDL EDITPROC clause.

Turning compression dictionary into load module format

Once the compression dictionary has been prepared using DSN1COMP, we next link-edit the object module into load module format. Example 14-4 shows some sample JCL used to perform this activity.

Example 14-4 Link-edit dictionary object deck

```

//*****
//* LINK HDC DICTIONARY OBJECT INTO TEST LIB - NAMED DICTTEP2
//*****
//LINK1    EXEC PGM=IEWL,
//          PARM='SIZE=(180K,28K),RENT,REFR,NCAL,LET,XREF,LIST=ALL'
//SYSLIN   DD DSN=PAOLOR5.DSNDBO4.TSDEPT.COMP.DICT,DISP=SHR
//SYSLMOD  DD DSN=PAOLOR5.ICSF.COMPDICT(DICTTEP2),DISP=SHR
//SYSUT1   DD UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,(10,1),RLSE)
//SYSPRINT DD SYSOUT=*
//*
```

In this sample, the SYSLIN DD statement identifies the sequential object deck created by DSN1COMP. The SYSLMOD DD statement names the partitioned data set used to store the linked compression dictionary (in this example named DICTTEP2).

Tip: In this step, and subsequent steps where we create link-edited artifacts, we chose to use separate libraries. While this helps in demonstrating the separate steps in the process, there is also a practical issue to consider. The sample JCL uses the DB2 SDSNEXIT as the target library for each of the different load modules created in the process. Because SDSNEXIT is an APF-authorized library, most customers are security sensitive to what elements are allowed into APF authorized libraries, including DB2 EDITPROCs. Our recommendation, from a security standpoint, is to link all the intermediate artifacts into non-APF-authorized PDSs, and only link the final version of the EDITPROC into SDSNEXIT.

Creating the compressing EDITPROC

The next step in the process is to prepare the compressing EDITPROC and include the link-edited compression dictionary into the EDITPROC. Example 14-5 on page 339 shows how to code the JCL to perform this step.

- ▶ The DICTLIB DD statement references the link-edited compression dictionary prepared in the previous step.
- ▶ SMPLIB DD points to the SMPE target library where Data Encryption for IMS and DB2 Databases Tool was installed
- ▶ SYSLMOD DD defines where the resultant compressing EDITPROC will be placed. This data set needs to be a load library formatted PDS.

It should have the following characteristics:

- Organization - PO
- Record format - U
- Record length - 0
- Block size - 32760

Care needs to be exercised in coding the link-edit control statements contained inside the SYSLIN control DD statement

- ▶ The CHANGE statement should reference your named link-edited dictionary. In our example, this is DICTTEP2. Do not replace the '(DSN1DICT)' part of the statement.
- ▶ The INCLUDE statement should also reference the named link-edited dictionary. In our example, this is DICTTEP2.
- ▶ The NAME statement will be the name you assign to the resultant compressing EDITPROC. For our purposes, we named this CPEPTEP2. This EDITPROC is placed into the library designated by SYSLMOD. In our example, this is PAOLOR5.ICSF.COMPRLIB.

Example 14-5 Linking compressing EDITPROC with dictionary

```
/******  
/* LINKEDIT COMPRESSION EDITPROC WITH DICTIONARY - NAMED CPEPTEP2  
/******  
//LINK2 EXEC PGM=IEWL,PARM='LIST,XREF,RENT,STORENX,NCAL'  
//SYSPRINT DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*  
//DICTLIB DD DSN=PAOLOR5.ICSF.COMPDICT,DISP=SHR  
//SMPLIB DD DSN=DEC.V1R1M0.SDECLMDO,DISP=SHR  
//SYSUT1 DD UNIT=SYSDA,SPACE=(1024,(50,50))  
//SYSLMOD DD DSN=PAOLOR5.ICSF.COMPRLIB,DISP=SHR  
//SYSLIN DD *  
CHANGE DICTTEP2(DSN1DICT)  
INCLUDE DICTLIB(DICTTEP2)  
INCLUDE SMPLIB(DECZLDX0)  
PAGE DSN1DICT  
ENTRY DECZLDX0  
NAME CPEPTEP2(R)  
/*  
/*
```

At this point you have created a functioning compression routine. In theory, you could implement this as an EDITPROC and you would be able to perform compression within your EDITPROC. For our purposes, we need to prepare our encrypting EDITPROC, and link-edit both the compressing and encrypting editproc together.

Creating the encrypting EDITPROC

This process is identical to the earlier depiction of encrypting EDITPROC preparation. For this discussion we will provide some JCL (shown in Example 14-6 on page 340). For simplification and clarity, we created another PDS to store the link-edited encrypting EDITPROC. In this sample JCL, this is the DD statement SYSLMOD.

Example 14-6 Preparing the encrypting EDITPROC

```
/* *****  
/* LINKEDIT EDITPROCS INTO THE USER EXIT DRIVER - ENCRDEP2  
/* *****  
//LINK3 EXEC PGM=IEWL,PARM='LIST,XREF,RENT'  
//SYSPRINT DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*  
//SDECLMDO DD DSN=DEC.V1R1M0.SDECLMDO,DISP=SHR  
//SCSFMOD0 DD DSN=CSF.SCSFMODO,DISP=SHR  
//SYSUT1 DD UNIT=SYSDA,SPACE=(1024,(50,50))  
//SYSLMOD DD DSN=PAOLOR5.ICSF.ENCRPLIB(ENCRDEP2),DISP=SHR  
//SYSLIN DD *  
ENTRY DECENA00  
INCLUDE SDECLMDO(DECENA00)  
INCLUDE SCSFMODO(CSNBKRR)  
NAME ENCRDEP2(R)  
/*  
/*  
//BATHTSO EXEC PGM=IKJEFT01,DYNAMNBR=25,REGION=OM,COND=EVEN  
//SYSLIB DD DISP=SHR,DSN=PAOLOR5.ICSF.ENCRPLIB  
//ISPLLIB DD DISP=SHR,DSN=SYS1.LINKLIB  
//ISPPLIB DD DISP=SHR,DSN=ISP.SISPPENU  
//ISPSLIB DD DISP=SHR,DSN=ISP.SISPSENU  
//ISPMLIB DD DISP=SHR,DSN=ISP.SISPMENU  
//ISPTLIB DD DISP=SHR,DSN=ISP.SISPTENU  
//SYSPROC DD DISP=SHR,DSN=ISP.SISPCLIB  
//SYSEXEC DD DISP=SHR,DSN=ISP.SISPEXEC  
// DD DISP=SHR,DSN=DEC.V1R1M0.SDECCEXE  
//ISPTABL DD DISP=SHR,DSN=PAOLOR5.SC63.ICSF.ISPPROF  
//ISPPROF DD DISP=SHR,DSN=PAOLOR5.SC63.ICSF.ISPPROF  
//SYSTSPRT DD SYSOUT=*  
//ISPLOG DD SYSOUT=*,DCB=(BLKSIZE=800,LRECL=80,RECFM=FB)  
//SYSTSIN DD *  
PROFILE PREFIX(PAOLOR5)  
ISPSTART CMD(%DECENC02 DB2 ENCRDEP2 -  
CLEAR.KEY.FOR.ENCRYPT.TABLE )  
/*
```

For review, we have created the following artifacts and stored them in individual non-APF authorized load libraries:

- ▶ The link-edited compressing dictionary, called DICTTEP2 and stored in COMPDICT
- ▶ The link-edited compressing EDITPROC, called CPEPTEP2 and stored in COMPLIB
- ▶ The link-edited encrypting EDITPROC, called ENCRDEP2 and stored in ENCRPLB

Creating the encrypting and compressing EDITPROC

The final step in the process is to combine all of these together into a single EDITPROC that will both compress and encrypt.

As mentioned before, this will be the name declared in the EDITPROC clause of the CREATE table statement, and as such needs to be link-edited into an APF authorized library (in our example, SDSNEXIT).

As shown in Example 14-7, this last step combines the compressing and encrypting EDITPROCS into a link-edited EDITPROC. In this example the following statements are true:

- ▶ ENCRYLIB DD points to the library that contains the encrypting EDITPROC ENCRDEP2.
- ▶ COMPLIB DD points to the library that contains the compressing EDITPROC CPEPTEP2.
- ▶ SYSLMOD DD points to the SDSNEXIT library where our combined EDITPROC will be placed.

Care needs to be taken in changing the link-edit control statements in the SYSLIN DD statements. In this sample, the INCLUDE statements after the SMPLIB statements should correspondingly reference the names of the compressing and encrypting EDITPROC.

Attention: This example is for creating a clear key EDITPROC. If you need to prepare a secure key EDITPROC, there would be different INCLUDE statements required

Example 14-7 Linking the compressing and encrypting EDITPROCs

```
//LINK4 EXEC PGM=IEWL,PARM='LIST,XREF,RENT,STORENX'
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//ENCRYLIB DD DSN=PAOLOR5.ICSF.ENCRPLIB,DISP=SHR
//SMPLIB DD DSN=DEC.V1R1M0.SDECLMDO,DISP=SHR
//COMPLIB DD DSN=PAOLOR5.ICSF.COMPLIB,DISP=SHR
//SYSUT1 DD UNIT=SYSDA,SPACE=(1024,(50,50))
//SYSLMOD DD DSN=DB9A9.SDSNEXIT(DREPDEP2),DISP=SHR
//SYSLIN DD *
REPLACE COMPEXIT(DECZLDX0)
REPLACE ENCREXIT(DECENA00)
INCLUDE SMPLIB(DECENADV)
INCLUDE COMPLIB(CPEPTEP2)
INCLUDE ENCRYLIB(ENCRDEP2)
PAGE DSN1DICT
ENTRY DECENADV
NAME DREPDEP2(R)
/*
//
```

Use of the EDITPROC prepared in the above fashion would first call the compression EDITPROC to compress the row, then call the encryption EDITPROC to perform the encryption process.

14.3.2 Additional encryption considerations with compressed data

When implementing compression and encryption using the Data Encryption for IMS and DB2 Databases Tool, the recommendation is that the COMPRESS YES attribute be removed from the TABLESPACE declaration. In addition, the REORG and LOAD control statements should have the KEEPDICTIONARY parameters removed.

With an encrypted table, if there are changes in the characteristics in the data after the creation of the external compression dictionary, there may be a requirement to build a new compression dictionary, and, by default, a new compressing EDITPROC. To perform this task, the following procedure needs to be performed.

1. UNLOAD the existing table data, this will create uncompressed and unencrypted data. Save this UNLOAD SYSREC and SYSPUNCH as this will allow you to avoid a second UNLOAD further in the process

2. DROP and CREATE the table without an EDITPROC specification. Remember, this is a destructive alteration, and you must capture all of the associated elements before the DROP, to recreate all of the table elements during the CREATE.
3. LOAD the unloaded data, this will give you an unencrypted and uncompressed tablespace. This is a requirement to run DSN1COMP and to allow it to build a usable compression dictionary and to compute an accurate compression prediction.
4. Prepare the new encrypting/compressing EDITPROC.

Restriction: If the newly recreated EDITPROC is identically named as an earlier used EDITPROC, a recycle of the DB2 address space will be necessary to get the newly prepared EDITPROC loaded into DB2 storage

5. A second DROP and CREATE is necessary to include the EDITPROC declaration on the table CREATE.
6. LOAD the unloaded data, using the retained SYSREC and SYSPUNCH from the earlier UNLOAD.

This scenario will result in compressed and encrypted data using the new compression dictionary.

Attention: Due to the complexity of dealing with the multiple drop scenario when rebuilding the compression dictionary, tables with data characteristics that require frequent rebuilding of the compression dictionary are not good candidates for compression with Data Encryption for IMS and DB2 Databases Tool. These tables should be encrypted only.

14.3.3 Compression scenario

Our workload generated tables consisted of predominately numeric columns and did not exhibit significant opportunities for compression benefits. We used another source for our compression scenario that was more typical of the type of data one would expect would compress well.

We first ran a series of tests with an uncompressed and unencrypted version of this table, and captured some space and page utilization numbers from the DB2 Administration Tool Space Management Dialog. The results of this are shown in Figure 14-25 on page 342. Of particular interest from a compression standpoint is the Active pages value.

```

DB2 statistical data: RUNSTATS timestamp: 2009-03-03-17.30.49.660503
Active pages: 22318          Pct. active : 57          Far reloc. : 0
Rows          : 42000        Pct. dropped: 0          Near reloc. : 0
                                   Compr. save : 0

Alloc   (KB): 97248

VSAM catalog information for: DB9AU.DSNDBC.A130X997.LIPS.I0001.A001
Alloc type : TRACK          First UNIT : 3390        Extents   : 11
Prim. alloc : 1             High alloc : 97248       Primary (KB): 48
Secd. alloc : 15            High use(KB): 89272     Second. (KB): 720
Percent used: 92
Volsers    : SBOX0G

```

Figure 14-25 Uncompressed sample tablespace - DB2 Administration Tool

Next, we prepared the encrypting and compressing EDITPROC as described above. When we ran the DSN1COMP step to create the external compression dictionary, Figure 14-26 shows the output from this execution, with the interesting values for compression savings highlighted

```

DSN1999I START OF DSN1COMP FOR JOB PAOLOR5L BUILD
DSN1998I INPUT DSNAME = DB9AU.DSNDBC.A130X997.LIPS.I0001.A001      , VSAM
DSN1944I DSN1COMP INPUT PARAMETERS
          4,096 DICTIONARY SIZE USED
              0 FREEPAGE VALUE USED
              5 PCTFREE VALUE USED
          99,999 ROWLIMIT REQUESTED
              ESTIMATE BASED ON DB2 LOAD METHOD
              255 MAXROWS VALUE USED
          DICTTEP4 EXTERNAL DICTIONARY OBJECT NAME

DSN1940I DSN1COMP COMPRESSION REPORT
          51,134 KB WITHOUT COMPRESSION
          10,231 KB WITH COMPRESSION
              79 PERCENT OF THE BYTES WOULD BE SAVED

              199 ROWS SCANNED TO BUILD DICTIONARY
          42,000 ROWS SCANNED TO PROVIDE COMPRESSION ESTIMATE
          4,096 DICTIONARY ENTRIES

              1,249 BYTES FOR AVERAGE UNCOMPRESSED ROW LENGTH
              252 BYTES FOR AVERAGE COMPRESSED ROW LENGTH

              28 DICTIONARY PAGES REQUIRED
          14,000 PAGES REQUIRED WITHOUT COMPRESSION
          2,828 PAGES REQUIRED WITH COMPRESSION
              79 PERCENT OF THE DB2 DATA PAGES WOULD BE SAVED

DSN1937I DSN1COMP TXT-DECK DICTTEP4 BUILT          1,173 RECORDS WRITTEN
DSN1994I DSN1COMP COMPLETED SUCCESSFULLY,        22,318 PAGES PROCESSED

```

Figure 14-26 DSN1COMP Compression Dictionary built output

We next dropped and recreated the table schema, with the EDITPROC specified in the new version of the table. We executed the load utility, ran RUNSTATS, and using the DB2 Administration Tool Space Management facility, captured space information as shown in Figure 14-27. As shown in the highlighted section of the report, the EDITPROC-based compression resulted in 3467 pages being used to store the compressed pages. When compared to the original uncompressed page count of 22318, shown in Figure 14-26 on page 343, there was a compression saving of approximately 84%.

```

DB2 statistical data: RUNSTATS timestamp: 2009-03-03-17.34.05.167089
Active pages: 3467          Pct. active : 72          Far reloc. : 0
Rows          : 42000        Pct. dropped: 0          Near reloc. : 0
                                   Compr. save : 0

Alloc   (KB): 14448

VSAM catalog information for: DB9AU.DSNDBC.A130X997.LIPS.I0001.A001
Alloc type : TRACK          First UNIT : 3390        Extents   : 4
Prim. alloc : 1             High alloc : 14448       Primary (KB): 48
Secd. alloc : 15           High use(KB): 13868     Second. (KB): 720
Percent used: 96
Volsers    : SB0X0F

```

Figure 14-27 Editproc compressed table - DB2 Administration Tool

The next question was how did the EDITPROC based compression compare with the DB2 compression as implemented using the COMPRESS YES operand on the CREATE TABLESPACE statement. We dropped and recreated the object a third time. This time we included the COMPRESS clause on the CREATE TABLESPACE, and removed the EDITPROC. We then ran another LOAD operation and took a look at the status of the owning tablespace using the DB2 Administration Tool Space Management feature as shown in Figure 14-28. As noted, the EDITPROC compression was incrementally more efficient than the native DB2 implemented compression. These results are not statistically different, however, so we will say that the two approaches are equivalent in efficiency.

```

DB2 statistical data: RUNSTATS timestamp: 2009-03-03-17.40.17.787228
Active pages: 3595          Pct. active : 71          Far reloc. : 0
Rows          : 42000        Pct. dropped: 0          Near reloc. : 0
                                   Compr. save : 79

Alloc   (KB): 19488

VSAM catalog information for: DB9AU.DSNDBC.A130X997.LIPS.I0001.A001
Alloc type : TRACK          First UNIT : 3390        Extents   : 5
Prim. alloc : 1             High alloc : 19488       Primary (KB): 48
Secd. alloc : 15           High use(KB): 14380     Second. (KB): 720
Percent used: 74
Volsers    : SB0X0F

```

Figure 14-28 DB2 COMPRESS YES table - DB2 Administration Tool



Administration of encrypted objects

In this chapter we provide considerations related to administering and operating a DB2 environment with encrypted objects. We focus on the DB2 specific areas affected by an encryption implementation using the Data Encryption for IMS and DB2 Databases Tool.

This chapter contains the following sections:

- ▶ Backup and recovery (local site considerations)
- ▶ Disaster recovery considerations
- ▶ Key rotation
- ▶ Alteration of encrypted table schema
- ▶ Failure scenarios
- ▶ Performance measurements

15.1 Backup and recovery (local site considerations)

In general, normal local site backup and recovery processes should remain relatively unaffected by an encryption implementation. However, if your recovery approach involves the use of DSN1COPY or other such mechanisms that deal with the linear VSAM data set or output of the DB2 COPY utility outside of RECOVER, then there might be some considerations.

Care needs to be taken to ensure that the same keylabel and editproc that was used when the image copy was taken is in effect at the time the RECOVER utility is executed. Whenever a new key is introduced, as part of a key rotation exercise, you need to make sure that there is a point of consistent recovery before and immediately after the key rotation occurs. Our recommendation is as follows:

- ▶ Establish a quiesce point for consistency
- ▶ Create a FULL SHRLEVEL reference copy
- ▶ Perform the key rotation exercise
- ▶ Establish a second point of consistency
- ▶ Create a second FULL SHRLEVEL reference copy

If the key rotation ceremony included a DROP, then the rows associated with the image copy taking prior to the key rotation will no longer be in the catalog. If after the key rotation ceremony is complete, there is a requirement to fall back to a point in time, you need to ensure that the appropriate EDITPROC and key label are defined as part of the table schema. This could involve a second DROP, which would then make the object unrecoverable, as there would be no COPY rows recorded in SYSCOPY.

It becomes important to understand recovery ramifications that span across key rotations, and in the event of a fallback, might require the use of a mechanism other than RECOVER, such as DSN1COPY, to restore the data successfully.

Important: Once the pre key rotation COPY utility has been run, record the image copy information stored in SYSCOPY, and the OBJECT identifiers from the catalog definition of the encrypted table. In the event of a fallback from a subsequent key rotation, your only recovery option will involve the use of DSN1COPY.

15.2 Disaster recovery considerations

From a disaster recovery perspective, the planning for a recovery site exercise requires that the considerations detailed in the following sections are taken into account.

Hardware symmetry

Ensure that the same type of hardware and operating system support is available at the recovery site. This is important in an environment where you have a CEX2C-encrypted CKDS and require API support delivered through the use of CEX2C enabled services. This includes the use of clear key encryption in ICSF environments prior to HCR7751, and any use of the Secure Key API.

The easiest configuration is to ensure that the CEX2C environment installed at the disaster recovery location is at the same level of microcode, and contains the same number of features and cryptographic coprocessors, and is the same System z processor class.

If you are running on a z9 or z10, there could be some issues if the hosting machine at disaster recovery is an earlier generation machine. This is potentially an issue with some organizations who use one set of hardware for hosting production workloads, and then have a second set of machines, which for cost purposes are an earlier generation processor model that are used for both development activities. These sysplexes are not collocated and have secondary utilization as the failover site in the event of a disaster. In this scenario, planning needs to occur to ensure that the cryptographic support elements do not become incompatible due to backleveled hardware.

In general, when using a z10 at the local site, and falling back to a z9 at the recovery site, if both environments have CEX2C features installed, there should be few compatibility issues, with the following caveat. If you have implemented AES 192- or 256-bit encryption on the z10, note that these are not supported on the z9 (or z890/z990, if that is what is being used at the DR site). There is no way to decrypt with the AES 128 support at the recovery site, so if this configuration is anticipated for disaster recovery purposes, you should consider only running AES 128 until you can ensure that a z10 with the proper level of AES support can be made available at the disaster recovery location.

If the recovery site is running z990 and has the PCI X Cryptographic Coprocessor (PCIXCC) it should also be compatible with z9 or z10 and CEX2C. PCIXCC an asynchronous cryptographic coprocessor. It is a replacement for the Cryptographic Coprocessor Feature and PCI Cryptographic Coprocessor available on earlier generations of zSeries hardware. It is only available in a supported manner on a IBM e-Server zSeries 890 or 990.

Any earlier generation zSeries processor, z800, z900, or older, are not compatible with z9 or z10 and CEX2C. Discuss the encryption hardware requirements with your contingency planning officer to ensure that contracts with disaster recovery hosting locations include language that spells out the specific hardware configuration needed for encryption support.

Recovery assets

As we mentioned earlier, the CKDS contains all of the generated data encrypting keys and the DES Master Key verification pattern used to validate the hardware registers at ICSF startup. As such, the CKDS should be deemed a system critical file, and should be backed up with the same frequency as other critical z/OS system data sets, such as SYS1.PARMLIB. Many customers place these critical data sets on a special volume, designated as the SYSRES volume. This data set has special backup and recovery requirements. The CKDS should be placed on a volume with similar SYSRES backup and recovery characteristics.

Of equal importance to the use of Data Encryption for IMS and DB2 Databases Tool is the DB2-authorized program list data set commonly referred to as the DB2 SDSNEXIT library. This is the library where all of the Data Encryption for IMS and DB2 Databases Tool prepared EDITPROCs are link-edited into. This data set is critical to proper DB2 functioning and should be backed up on a frequent basis, and be available at the DB2 recovery site.

Master Key Entry at recovery site

Once the floor system at disaster recovery has been successfully initialized, and the ICSF CKDS data set has been restored, the first execution of ICSF will result in an environment with all cryptographic functions requiring a CEX2C-supported element disabled. This is similar to the scenario described earlier in the ICSF Master Key entry scenario. So, at the disaster recovery site, there will need to be a designated key officer who can run the ICSF Master Key entry dialog to enter the same master key used to initialize the CKDS. No new initialization of CKDS is necessary. Once the master key registers on the recovery site, CEX2C are the same as the local site contents, and the validation pattern of the CEX2C register will match the materials in the CKDS. At that point, all cryptographic functions supported by the CEX2C can be exploited.

Disaster recovery testing clean-up

An additional consideration regarding DES master key entry scenario being performed for a disaster recovery test. Once the CEX2C hardware registers have been loaded, they contain the DES master key used in your live production environment. Once the disaster recovery test has concluded, one of the steps in the clean-up process needs to include a procedure to ensure that these valid MK register contents be removed and replaced with bogus key values.

To do this, you should prepare two sets of previously generated random numbers that were generated at your local site. Once the disaster recovery testing has been concluded, the key officer should perform two separate key rotation exercises to load invalid keys into the hardware. If you recall, there are three sets of keys stored in the hardware registers: one is the current key value, one is the new key values, and one is the old key value. The contents of the the registers are shown in Figure 15-1.

```
. ----- ICSF - Coprocessor Hardware Status ----- .
.  COMMAND ==>                                     SCROLL ==> .
.                                                    CRYPTO DOMAIN: 2 .
.
. REGISTER STATUS                                COPROCESSOR E01 .
.                                                    More:      + .
.  Crypto Serial Number      : 94000264 .
.  Status                    : ACTIVE .
.  DES Master Key
.    New Master Key register  : EMPTY .
.    Verification pattern    : .
.    Hash pattern            : .
.    : .
.  Old Master Key register    : EMPTY .
.    Verification pattern    : .
.    Hash pattern            : .
.    : .
.  Current Master Key register : VALID .
.    Verification pattern    : F070C0451E30EC3F .
.    Hash pattern            : 2100222009E844DC .
.    : A00C7DA63F29B26C .
.  AES Master Key
.    New Master Key register  : NOT SUPPORTED .
.
. Press ENTER to refresh the hardware status display. .
. Press END   to exit to the previous menu. .
.
```

Figure 15-1 ICSF Coprocessor Hardware Status

When the original key value is entered at the start of the disaster recovery test, the Current Master Key register has data, the other two sets of registers are empty or in an unknown state (possibly the master keys or dummy master keys) from the previous customer's DR exercise. At the conclusion of the disaster recovery test, when the first bogus key is entered, the New Master Key register contains the bogus key values. When the first bogus key is loaded, the Current Master Key register now contains the bogus key, and the valid or real key values are stored in the Old Master Key register.

You next enter a second bogus key, which is placed into the New Master Key register. When this key is loaded, the second bogus key now becomes the Current Master key, and the contents of the Old Master Key register, which had held the valid key values, is now replaced with the value of the first bogus key. At this point, all of the register contents are bogus, and any subsequent viewing of the contents of the registers will not constitute an exposure of a valid master key value.

15.3 Key rotation

Key rotation is a concept where at some pre-determined interval, keys used to encrypt data have to be changed. Some regulatory compliance initiatives specifically call this out as a set period of time, for example every 12 months. Other times, key rotation needs to be performed whenever a breach or key compromise, whether actual or suspected, has occurred.

There are several different interpretations of this requirement. One is that in an environment where data encrypting keys are stored in a CEX2C protected CKDS, rotation of the DES master key is sufficient. In this scenario, the data encrypting key does not change in value, but the key as stored on the CKDS is encrypted with a different DES master key. This key rotation exercise is performed by the ICSF administrator, can be performed in a nondisruptive manner, and does not require an unload of the data or any type of outage.

The second interpretation of this requirement calls for the data that is protected by the original key, to be re-encrypted by a new key value. For this type of rotation to occur, the data must be first unloaded under the old key, then reloaded using the new key. If you recall, if you need to make a change to an existing (and loaded into DB2 memory) EDITPROC, a bounce of DB2 will be required to get the new copy of the EDITPROC loaded into DB2 storage.

In the first approach, to avoid this DB2 outage to get an EDITPROC refresh, whenever a data encrypting key needs to be retired or changed, the technique should be as follows:

1. UNLOAD using the existing (old key) EDITPROC
2. Create a new key with a unique keylabel and rebuild EDITPROC pointing to new keylabel
3. DROP/RECREATE the table and specify the new EDITPROC
4. LOAD the table, this will encrypt the data under the new key.

There is also another flavor of the process described above, which would require a DB2 outage, but could avoid the DROP/CREATE requirement. This approach would be as follows:

1. UNLOAD using the existing (old key) EDITPROC pair
2. Create a new key with a unique keylabel and rebuild the existing EDITPROC pointing to new keylabel
3. Bounce DB2, this will ensure that the new version of the existing EDITPROC is loaded when next referenced
4. LOAD the table, this will invoke the EDITPROC with the new keylabel

In some environments, where the key rotation process could be scheduled in conjunction with a planned outage of DB2, the second approach might be a little less disruptive.

Obviously the second approach is only for a customer who is concerned about the downstream ramifications of the DROP, mainly the loss and subsequent requirement to recreate elements such as views, synonyms, authorizations, and other related items that are lost on the DROP. In addition, if the requirement includes the ability to support the use of COPY data sets encrypted under the old key, but after a key rotation exercise has occurred, the recommendation would be to leave the original key and EDITPROC intact, and create a new EDITPROC and keylabel pair as described in the first scenario.

Key retention and retirement

Once a key rotation policy and approach has been defined and implemented, the issue of key retention and retirement needs to be addressed. As we discussed in the previous section on recovery, there are certainly local site and remote site considerations that need to be considered.

In addition, key retention and retirement has some effect on your ability to perform archiving of inactive data for long term retention. In general, if the archive process is driven from using a utility like UNLOAD to create the archive file, then the encryption process and retention of keys and EDITPROCs is generally not an issue. Once the archive file is created, and is in cleartext as a result of the decryption performed by the Data Encryption for IMS and DB2 Databases Tool, then it is completely usable. One possible exception would be that in many situations, along with the unloaded archive file, the DDL that describes the table schema in effect at the time of the archive is also saved. If this is the case, you would either need to modify this saved DDL to remove the EDITPROC declarative in the event of some future archive file retrieval, or retain the original EDITPROC as stored in the SDSNEXIT library along with the original Key and Keylabel pair stored in the CKDS.

If you adopt a policy that ensures key and editproc retention for long periods of time, any time a key rotation exercise is performed a new EDITPROC and Key are created. This should ensure that the necessary crypto artifacts are available for future archive retrieval requests.

Tip: Our general recommendation, where possible, is to retain old keys and EDITPROCs in anticipation of future archive retrieval requirements

Some regulations stipulate specific requirements for key retention that may conflict with our recommendation to retain old keys and EDITPROCs. Confer with your security and privacy officers to determine the appropriate key retention policy that fits particular needs.

15.4 Alteration of encrypted table schema

There are some restrictions on altering a table schema if an EDITPROC has been defined. We discuss the alternatives in this section.

EDITPROC restrictions

An edit routine receives an entire table row, and can transform that row in any way. Also, it receives a transformed row and must change the row back to its original form. You must not specify an edit routine for a table with a LOB, ROWID, identity column, or security label column.

Your edit routine can encode the entire row of the table, including any index keys. However, index keys are extracted from the row before the encoding is done. Therefore, index keys are stored in the index in edit-decoded form. Hence, for a table with an edit routine, index keys in the table are edit-coded. Index keys in the index are not edit-coded.

An edit routine is not invoked for a DELETE operation without a WHERE clause that deletes an entire table in a segmented table space.

You cannot use ALTER TABLE to change any characteristics of a table where an EDITPROC has been defined. To demonstrate this restriction, we used the DB2 Administration Tool to alter/add a simple column to an existing table with an EDITPROC assigned. Figure 15-2 on page 351 shows the initial panel in the ALTER TABLE dialog.

```

DB2 Admin ----- DB9A Alter Table ----- 12:31
Command ==>

Table owner ==> PAOLOR5 >
Table name  ==> GLWTEPA      >

AUDIT          ==> ALL      (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

s 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 15-2 Administration Tool Alter Table

After specifying the ADD column option, we use the ALTER TABLE dialog to create a new column, NEWCOL, which is CHAR column, and 4 bytes in length. Figure 15-3 shows this dialog.

```

DB2 Admin ----- DB9A Alter Table ----- 12:34
Command ==>

ALTER TABLE
Table owner ==> PAOLR5 >
Table name ==> GLWTEPA >
ADD
Column name ==> newcol >
Column type ==> char Built-in only
Column leng ==> 4 " " "
Precision ==> " " " (opt, w/FLOAT, DECIMAL)
Scale ==> " " " (opt, w/DECIMAL)
Schema name ==> > User-defined only
Data type ==> > " " " "

Allow Nulls ==> (Yes-nullable/No-NOT NULL)
FOR ? DATA ==> (B-Bit, S-SBCS, M-Mixed, blank-N/A)
W. DEFAULT ==> (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 15-3 DB2 Administration Tool - Add Column dialog

Upon pressing enter, the ALTER TABLE statement built from the DB2 Administration Tool dialog is executed, with the resulting SQL code raised by DB2 based on the restrictions of ALTER on tables with EDITPROCs. Figure 15-4 shows the resultant SQL code.

```
DB2 Admin ----- DB2 Error Display 1 -----
12:38
Command ==>
Rollback done
      SQLCODE : -668                      DSNTIAR CODE : 0

DSNT408I SQLCODE = -668, ERROR:  THE COLUMN CANNOT BE ADDED TO THE TABLE
      BECAUSE THE TABLE HAS AN EDIT PROCEDURE
DSNT418I SQLSTATE  = 56018 SQLSTATE RETURN CODE
DSNT415I SQLERRP   = DSNXIAB3 SQL PROCEDURE DETECTING ERROR
DSNT416I SQLERRD   = 130 0 0 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD   = X'00000082' X'00000000' X'00000000' X'FFFFFFFF'
      X'00000000' X'00000000' SQL DIAGNOSTIC INFORMATION
```

Figure 15-4 SQL Code -668 raised by ALTER TABLE

In lieu of ALTER TABLE, you will need to perform DROP/CREATE-based table alteration.

- ▶ Run the UNLOAD utility or run the REORG TABLESPACE utility with the UNLOAD EXTERNAL option to unload the data and decode it using the existing edit procedure or field procedure. These utilities generate a LOAD statement in the data set (specified by the PUNCHDDN option of the REORG TABLESPACE utility) that you can use to reload the data into the original table space. For a table space in which the maximum record length is greater than 32 KB, use the DSNTIAUL sample program to unload the data
- ▶ Using an administration product such as DB2 Administration Tool, or manually generated SQL, collect all the information about the existing table, and all of the related objects and elements. This information will be used to recreate the table characteristics after the DROP.
- ▶ DROP the table.
- ▶ Recreate the table using the information about the table and related objects obtained from the catalog prior to the DROP, and apply any necessary changes. Modify the code of the edit procedure and rebuild the EDITPROC.
- ▶ Use the LOAD utility to reload the data.
- ▶ The DROP may have invalidated packages or plans that reference the table. You might need to rebind these packages and plans.

Attention: The ALTER restriction for objects with high availability requirements might need be considered when planning an encryption implementation

Triggers and encrypted tables

Triggers (both BEFORE and AFTER) work at the SQL level, not at the data level. Therefore, because SQL is always working with decoded data, the triggers would too. Hence, the data that is being manipulated by the triggers would be in its unencrypted text format.

Based on the above information, you may need to consider the sensitivity of the data that is being manipulated by the triggers. If it still needs to be encrypted, you may need to consider adding encryption to the target table.

RACF

You can use z/OS Security Server RACF to control which applications can use specific keys and services. This can help you ensure that keys and services are used only by authorized users and jobs. You can also use RACF to audit the use of keys and services. The XCSFKEY class controls who can export a token using the Symmetric Key Export callable service (CSNDSYX). To set up these controls, you create and maintain RACF general resource profiles in the CSFKEYS class, the CSFSERV class and the XFACILIT class. The CSFKEYS class controls access to cryptographic keys with the key label, the CSFSERV class controls access to ICSF services, and resources in the XFACILIT class define a key store policy that controls the use of key tokens that are stored in the CKDS and PKDS.

To set up profiles in the CSFKEYS general resource class, take the following steps:

1. Define appropriate profiles in the CSFKEYS class: `RDEFINE CSFKEYS label UACC(NONE) other-optional-operands` where label is the label by which the key is defined in the CKDS or PKDS (this is not the transport key label). Note that if an application uses a token instead of a key label, no authorization checking is done on the use of the key.

Notes:

- If you have ICSF/MVS Version 1 Release 1 profiles that specify `key-type.label`, you need to change them to specify only label.
- As with any RACF profile, if you want to change the profile later, use the `RALTER` command. To change the access list, use the `PERMIT` command as described in the next step.
- If you have already started ICSF, you need to refresh the in-storage profiles. See Step 3. You can specify other operands, such as auditing (AUDIT operand), on the `RDEFINE` or `RALTER` commands.
- If the RACF security administrator has activated generic profile checking for the CSFKEYS class, you can create generic profiles using the generic characters `*` and `%`. This is the same as any RACF general resource class.

2. Give appropriate users (preferably groups) access to the profiles:

```
PERMIT profile-name CLASS(CSFKEYS) ID(groupid) ACCESS(READ)
```

3. When the profiles are ready to be used, ask the RACF security administrator to activate the CSFKEYS class and refresh the in-storage RACF profiles:

```
SETROPTS CLASSACT(CSFKEYS) SETROPTS RACLIST(CSFKEYS) REFRESH
```

4. By default, when CSFKEYS is active, all problem state callers of ICSF services using a key resources protected by CSFKEYS are checked by RACF. By default, authorized requestors or not checked. To enable CSFKEYS checking of supervisor state or authorized programs require the ICSF parameter `AUTHCHECK` to be set to `YES`.

In a DB2 environment, we suggest that `AUTHCHECK` be turned to `NO`. The reason for this is due to the nature of the way that SQL statement processing in DB2 runs under the SQL requestors TCB.

We are using the keylabel CLEAR.KEY.FOR.ENCRYPT.TABLE that was originally associated with the EDITPROC CLEAREXT. Using this EDITPROC, we successfully load and encrypt our table.

At some subsequent point, we relink the EDITPROC and specify a different keylabel, as shown in Figure 15-6.

```

ISPSTART CMD(%DECENC02 DB2 CLEAREXT  SG24.7720.00.CLEAR.KEY.01
)
DEC140I ICSF key successfully implemented for:  DBMS = DB2 , KEY LABEL =
SG24.7720.00.CLEAR.KEY.01 .

*****
DYNAMICALLY INVOKED ZAP SYSIN CONTROL STATEMENTS FOLLOW
*****

    NAME  CLEAREXT
    VER   0101 E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7
    REP   0101 E2C7F2F44BF7F7F2F04BF0F04BC3D3C5C1D94BD2C5E84BF0F1

*****
DYNAMICALLY INVOKED ZAP SYSPRINT OUTPUT FOLLOWS
*****

AMASPZAP  INSPECTS, MODIFIES, AND DUMPS CSECTS OR SPECIFIC DATA RECORDS ON DIRECT ACCESS
STORAGE.
    NAME  CLEAREXT
    VER   0101 E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7
    REP   0101 E2C7F2F44BF7F7F2F04BF0F04BC3D3C5C1D94BD2C5E84BF0F1
AMA122I OLD DATA WAS E7E7E7E7E7E7E7E7E7E7E7E74040404040404040404040404040404040404040
AMA125I CLEAREXT IDR COUNT = 0001 (MAX=0019)
AMA100I AMASPZAP PROCESSING COMPLETED
    PAOLOR5.PAOLOR5X.JOB25228.D0000106.? was preallocated (no free was done).

```

Figure 15-6 Updated key ZAP output

After the updated EDITPROC is rebuilt and gets loaded into DB2 storage, any subsequent SQL access will use a clear key value that is different from the one used to convert the data to ciphertext. As a result, the data will not be properly decrypted. And, because this is a valid clear key that is known to the CKDS, the encrypt instruction KMC will return data to the caller, (in this case DB2) but the data will be ciphertext and any subsequent access will result in unpredictable results. For our scenario, we recycled DB2 to force the new version of the EDITPROC to be loaded into DB2 storage. Then, using DSNTEP2, we attempted to access the encrypted table with the incorrect key, resulting in the behavior shown in Figure 15-7 on page 357.

```

READY
DSN SYSTEM(DB9A)
DSN
RUN PROGRAM(DSNTEP2) PLAN(DSNTEP2) LIB('DB9AU.RUNLIB.LOAD')
(2656,,)/ALIGN(MID)')
DSN ENDED DUE TO ERROR+
SYSTEM ABEND CODE 0C7 REASON CODE 00000000

PAGE 1
***INPUT STATEMENT:
SELECT * FROM PA0LOR5.NEWTEPA FETCH FIRST 10 ROWS ONLY
IBM0537S ONCODE=8097 Data exception

```

Figure 15-7 S0C7 abend using incorrect clear key

When DB2 is mapping the ciphertext into the individual columns of the row being returned to the SQL requestor, the invalid data represented in the ciphertext values resulted in S0C7 data exception. It is possible that if the table had been defined with all columns as character representation, the SQL statement might have succeeded, but the column data would be ciphertext and as a result useless.

15.5.2 CKDS failure - Master key mismatch

The next scenario attempts to cover the situation where there is a problem with a mismatch between the CKDS and the master key values stored in the CEX2C hardware register. To recreate this, we attempt to start ICSF up with a different CKDS that used a different DES master key for initialization.

Attention: ICSF can provide some minimal cryptographic functions without the presence of a CEX2C feature, so CKDS mismatch problems, while detected at ICSF startup, are not deemed fatal errors.

When ICSF starts up, there is a comparison made between the contents of the CKDS, the hash and verification patterns that are stored in the header record of the CKDS, and the hash and verification patterns stored in the hardware registers of the CEX2C. If these patterns match, the CKDS is deemed to be valid, and ICSF initializes without any compromise. If these patterns do not match, the mismatch is detected and noted in the ICSF started task joblog, and those ICSF features that require the use of the CEX2C are disabled.

In our scenario, we changed the ICSF parameters to point to a different CKDS data set, one that was not initialized under the current CEX2C master key. In looking at the started task joblog for ICSF, we can see the messages as shown in the Figure 15-8 on page 358.

```

STC25263 ---- SATURDAY, 21 FEB 2009 ----
STC25263 IEF695I START CSF      WITH JOBNAME CSF      IS ASSIGNED TO USER STC      ,
GROUP SYS1
STC25263 $HASP373 CSF      STARTED
STC25263 IEF403I CSF - STARTED - TIME=15.35.08 - ASID=0095 - SC63
STC25263 CSFM607I A CKDS KEY STORE POLICY IS NOT DEFINED.
STC25263 CSFM607I A PKDS KEY STORE POLICY IS NOT DEFINED.
STC25263 CSFM610I GRANULAR KEYLABEL ACCESS CONTROL IS DISABLED.
STC25263 CSFM611I XCSFKEY EXPORT CONTROL FOR AES IS DISABLED.
STC25263 CSFM611I XCSFKEY EXPORT CONTROL FOR DES IS DISABLED.
STC25263 CSFM123E MASTER KEY DES ON CRYPTO EXPRESS2 COPROCESSOR E01, SERIAL NUMBER
94000264, IN ERROR.
STC25263 CSFM012I NO ACCESS CONTROL AVAILABLE FOR CRYPTOZ RESOURCES. ICSF PKCS11
SERVICES DISABLED.
STC25263 *CSFM122I PKA SERVICES WERE NOT ENABLED DURING ICSF INITIALIZATION.
STC25263 CSFM001I ICSF INITIALIZATION COMPLETE
STC25263 CSFM126I CRYPTOGRAPHY - FULL CPU-BASED SERVICES ARE AVAILABLE.
STC25263 CSFM401I CRYPTOGRAPHY - SERVICES ARE NO LONGER AVAILABLE.

```

Figure 15-8 ICSF startup with CKDS master key mismatch

In our scenario, being an artificially created situation, we know why we experienced this error. With a similar situation occurring in a production environment, there is a technique that can be used to compare the hash and verification patterns to reconcile the differences between the CKDS and CEX2C values.

First, print the header record on the CKDS that is being accessed by the ICSF startup. The hash and verification pattern locations are documented in *z/OS Cryptographic Services ICSF System Programming Guide, SA22-7520*. The relevant documentation showing the offsets we are interested in is shown in Table 15-1.

Table 15-1 CKDS verification pattern offsets

Offset (dec)	# of bytes	Field name	Description
108	8	DES master key verification pattern	The system DES master key verification pattern.
116	8	DES master key authentication pattern	The system DES master key authentication pattern.
124	8	AES master key verification pattern	The AES master key verification pattern.
132	6	Reserved	The field is set to binary zeros.
196	52	Installation data	Installation data associated with the CKDS record, as supplied by an installation exit.
248	4	Authentication code	The code generated by the authentication process that ensures that the CKDS record has not been modified since the last update. The authentication code is placed in the CKDS header record when the CKDS is initialized. ICSF verifies the CKDS header record authentication code whenever a CKDS is re-enciphered, refreshed or converted from PCF to ICSF format. This field is not used when the record level authentication flag is set in the CKDS header flag bytes field of the CKDS header record.

Using any facility to print or view VSAM data sets (in our case we've elected to use IDCAMS print), view the first record on the CKDS. This is the CKDS header record. Record the values of the verification and authentication pattern stored in the CKDS header record. Figure 15-9 is an example of the header record from our system.

```

000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
000020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
000040 00000000 00000000 F2F0F0F9 F0F2F0F9 F1F9F0F4 F5F1F9F1 F2F0F0F9 F0F2F2F0
000060 F1F3F1F5 F4F7F5F9 00078080 F070C045 1E30EC3F 00000000 00000000 00000000
000080 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0000A0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0000C0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0000E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000

```

Figure 15-9 CKDS header record

Using the offset described in the *z/OS Cryptographic Services ICSF System Programming Guide*, SA22-7520, we see that the CKDS has a verification pattern of F070C0451E30EC3F. Next, look at the CEX2C master key verification pattern for the current active registers. To look at this information, access the Coprocessor Management functions of the ICSF administration dialogs. This was described in Chapter 12, “Architecture and ICSF key management” on page 277, so we just show the Coprocessor Hardware Status display in Figure 15-10.

```

. CSFCMP40 ----- ICSF - Coprocessor Hardware Status ----- .
. COMMAND ==>                                SCROLL ==> .
.                                           CRYPTO DOMAIN: 2 .
.                                           .
. REGISTER STATUS                            COPROCESSOR E01 .
.                                           More:      + .
.   Crypto Serial Number      : 94000264 .
.   Status                    : ACTIVE .
.   DES Master Key .
.     New Master Key register : EMPTY .
.     Verification pattern    : .
.     Hash pattern            : .
.     .                       : .
.     Old Master Key register : EMPTY .
.     Verification pattern    : .
.     Hash pattern            : .
.     .                       : .
.     Current Master Key register : VALID .
.     Verification pattern    : F070C0451E30EC3F .
.     Hash pattern            : 2100222009E844DC .
.     .                       : A00C7DA63F29B26C .
.   AES Master Key .
.     New Master Key register : NOT SUPPORTED .
.   .
. Press ENTER to refresh the hardware status display. .
. Press END   to exit to the previous menu. .
.

```

Figure 15-10 ICSF Coprocessor Hardware Status

Notice that the value from the CKDS header record and the register values displayed in the ICSF panel match. This process will verify that ICSF is running on the correct CKDS. If they were not matching, you would see symptoms similar to those documented in Figure 15-8 on page 358.

In the preceding scenario, it was noted that although ICSF will detect any verification pattern mismatch and report on this during ICSF address space startup, the ICSF environment will complete initialization, and some limited cryptographic functionality will still be possible. The other test that we explored was the impact on an application issuing SQL requests that were not supported by the ICSF configuration. We ran a simple DSNTEP2 test, using a valid EDITPROC. This is the resulting SQL code that was raised by the first SQL request against an encrypted table.

15.5.3 Out-of-synch key labels

One other scenario that was explored was where an EDITPROC was prepared with an incorrect keylabel specification. In our example, we ran the linkedit and subsequent zap with a keylabel not defined in the CKDS. As you may image, this is not checked until execution time, and when the first SQL statement is executed, the result is the SQLCODE -652 as shown in Figure 15-11.

```
PAGE 1
***INPUT STATEMENT:
  SELECT * FROM PAOLR5.NEWTEPA FETCH FIRST 10 ROWS ONLY
SQLERROR ON SELECT COMMAND, FETCH FUNCTION
RESULT OF SQL STATEMENT:
DSNT408I SQLCODE = -652, ERROR: VIOLATION OF INSTALLATION DEFINED EDIT OR VALIDATION PROCEDURE
CLEAREXT
DSNT418I SQLSTATE = 23506 SQLSTATE RETURN CODE
DSNT415I SQLERRP = DSNXRR SQL PROCEDURE DETECTING ERROR
DSNT416I SQLERRD = 102 13172749 0 13228485 -992411648 -993672445 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD = X'00000066' X'00C9000D' X'00000000' X'00C9D9C5' X'C4D90000' X'C4C5C303' SQL
DIAGNOSTIC
INFORMATION
SUCCESSFUL RETRIEVAL OF 0 ROW(S)
```

Figure 15-11 SQLCODE with CKDS mismatch

Once the EDITPROC is loaded by DB2, it remains in storage until the next time DB2 is recycled. From a best practices standpoint, ensure that the EDITPROC points to the proper valid keylabel. One way to manage this scenario would be to prepare the EDITPROC and link into a test environment first, and use the EDITPROC to encrypt a small test table. This approach will catch any preliminary problems related to ICSF or the improper use of keys prior to production implementation.

15.6 Performance measurements

Generally, it is dangerous to extrapolate performance measurements from the type of data and workload available to us. However, we wanted to gather some general performance metrics and make some generalized observations about performance given our limited capability for workload and object size.

What was performed was a series of benchmarks looking at two different types of workload. The first was running a battery of typical utilities against our workload objects, in an attempt to measure impact to utilities performance. The second scenario involved executing the workload generator against the pre-allocated objects, including the encrypted table, and measuring subsequent activity.

In the execution of the benchmarks, we had the traces active, as shown in Figure 15-12. In addition, for the utilities benchmarks, we ran with DIAGNOSIS (100, 101), which provides some additional performance measurements for each utility phase, and results in approximately an additional 10% overhead in utility execution.

```

RESPONSE=SC63
DSNW127I  -DB9A CURRENT TRACE ACTIVITY IS -
TNO TYPE  CLASS          DEST QUAL IFCID
01  STAT  01,03,04,05, SMF  NO
01          06
02  ACCTG 01             SMF  NO
03  ACCTG 01,02,03      SMF  NO
04  ACCTG 01,02,03,07, SMF  NO
04          08
05  STAT  01,03,04      SMF  NO
06  AUDIT 03             OP1  NO
07  AUDIT 01,02,07,08  OP1  NO  090,091
*****END OF DISPLAY TRACE SUMMARY DATA*****
DSN9022I  -DB9A DSNWVCM1 '-DISPLAY TRACE' NORMAL COMPLETION

```

Figure 15-12 Traces active on DB9A

Once this performance information was collected, we used the batch reports generated by IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS. The reports used were the ACCOUNTING TRACE and STATISTICS LONG reports.

15.6.1 Utilities

We created a series of utilities that were executed in the following types of encryption environments:

- ▶ Un-encrypted text. This established our baseline measurements
- ▶ Clear TDES
- ▶ Clear AES

We were primarily interested in those utilities that require the use of EDITPROC to encode and decode the individual rows. These include the following utilities:

- ▶ UNLOAD
 - No special performance parameters taken
- ▶ LOAD
 - LOG NO RESUME NO REPLACE ENFORCE NO
- ▶ REORG
 - SORTDATA. With three indexes we observed parallel index rebuild with six tasks and unconstrained by storage or CPU
- ▶ RUNSTATS
 - TABLE ALL

We ran COPY for each set of objects, mainly to provide a point of consistent recovery in the event of some unanticipated failure. To validate that it was unaffected by the presence of encrypted pages and log records, we ran the RECOVER utility against an encrypted table.

For each combination of algorithm and utility, we ran three separate but consecutive utility executions. This was done to ensure that we saw consistent results from execution to execution. In almost every case, we observed little difference within each group. This led us to conclude that we had a consistent environment, with little competition for processor or I/O resources.

Important: This section describes general performance characteristics as observed in our test scenarios. This data was generated for benchmark purposes and is not representative of realistic production data. We encourage you to perform performance measurements in your environment, using your data and under your realistic workload conditions.

Baseline measurement

We conducted the baseline measurement to establish performance norms and compared the rest of the workload against these numbers. All these numbers are expressed in seconds, given the size of the objects.

UNLOAD

For UNLOAD on an encrypted table, there is one decrypt operation for each row unloaded in the table. Table 15-2 shows the relevant performance metrics as captured by OMEGAMON PE.

Table 15-2 Performance measurements for UNLOAD

Mode	Class 1 Elapsed	Class 2 elapsed	Class 1 CPU	Class 2 CPU
Unencrypted	17.4297	6.4910	6.3729	6.0504
TDES Clear	22.4910	9.7493	9.4245	9.0969
AES Clear	15.9197	9.7497	9.3996	9.0855

Unload is performed as part of the initial implementation of encryption, and when the EDITPROC is declared. It is required as part of any ALTER, as the rules on table changes with EDITPROC dictate that a DROP/RECREATE be performed. Based on Table 15-2, we experienced on average about a 47% increase in Class 1 CPU.

LOAD

Load requires one encrypt call for each row loaded into the target table. In our examples, we captured the performance metrics shown in Table 15-3.

Table 15-3 Performance measurements for LOAD

Mode	Class 1 Elapsed	Class 2 Elapsed	Class 1 CPU	Class 2 CPU
Unencrypted	30.9814	13.0834	13.3388	7.2345
TDES Clear	34.5662	16.7731	17.0803	10.9916
AES Clear	26.7806	17.6951	16.8166	10.7253

These measurements show a 28% increase in Class 1 CPU time. We also noticed better elapsed times for AES encryption than those experienced without encryption. The reason for this might have been related to some external influence that affected the workload. We also would have expected to see a similar increased percentage to UNLOAD, as the decrypt and encrypt operations should exhibit similar performance characteristics.

REORG

This utility experiences the most impact, the UNLOAD and RELOAD phases each require a crypto call. Table 15-4 shows the results of the REORG workload.

Table 15-4 Performance measurement for REORG

Mode	Class 1 Elapsed	Class 2 elapsed	Class 1 CPU	Class 2 CPU
Unencrypted	15.5581	7.7031	10.5002	2.8769
TDES Clear	22.7278	15.2198	17.7589	10.2486
AES Clear	22.5829	14.7294	17.2830	9.8058P

The results of REORG indicate a 65% increase in Class 1 CPU.

RUNSTATS

The other online utility with significant impact due to encryption is RUNSTATS. The choice of RUNSTATS parameters, specifically those parameters such as TABLE ALL, dictate that the table rows be extracted and decrypted to gather table level statistics. See Table 15-5.

Table 15-5 Performance measurement for RUNSTATS

Mode	Class 1 Elapsed	Class 2 elapsed	Class 1 CPU	Class 2 CPU
Unencrypted	1.5956	1.5950	0.4132	0.4126
TDES Clear	4.2280	4.2274	3.8367	3.8362
AES Clear	4.0386	4.0381	3.6802	3.6797

RUNSTATS scenarios indicate approximately 700% increase in CPU.

15.6.2 SQL

We conducted a workload on both an encrypted and unencrypted tables. The workload consisted of reads and update random processing against a table space with 17 tables. Two stored procedures with 240,000 transactions, 1.4 million reads (selects) were completed. See Table 15-6.

Table 15-6 Performance measurement for SQL

Mode	Class 1 Elapsed	Class 2 elapsed	Class 1 CPU	Class 2 CPU
Unencrypted	5.0128	5.0122	3.5901	3.5897
AES Clear	5.0159	5.0154	4.1034	4.1030
TDES Secure	14.5064	14.5064	0.2895	0.2851

The elapsed time overhead due to encryption is generally negligible for random processing with clear key encryption. The CPU overhead is less than 15%. For massive sequential processing, we expect the overhead to be similar to the UNLOAD utility (see Table 15-3 on page 362). With secure key, the overhead for random processing is about 2.5 times in elapsed time. The CPU overhead is not directly measurable under DB2 accounting, because encryption is executed within the hardware boundaries of the CEX2C feature. We expect the overhead to be even higher for sequential processing with secure key, to the point of not currently being a viable solution.

Part 6



Appendixes

This part consists of the following two appendixes:

- ▶ Appendix A, “System topology and workload” on page 367
- ▶ Appendix B, “Sample configuration files for DB2 Audit Management Expert for z/OS” on page 383



A

System topology and workload

This appendix describes the environment set up for this project.

- ▶ A.1, “Hardware and software set up” on page 368 describes the components of our test environment.
- ▶ A.2, “DB2 workload” on page 368 describes our installation and customization of the DB2 Workload Generator V1.5, an IBM internal tool.

A.1 Hardware and software set up

We have used a sysplex complex in an LPAR with 2094 z9 with two shared CPs and 4 GB of real storage.

The OS level is z/OS V1R10.

The DB2 9 for z/OS has been updated at level D08100 at the start of our project. We have added more recent maintenance as listed in the book.

The subsystem ID is DB9A.

A.2 DB2 workload

We have used the IBM Workload Generator Version 1.5 (GLW V1.5). The purpose of this tool is to create and drive a workload on a DB2 for z/OS database. This tool was written to provide a simple way of creating and driving a substantial workload on a DB2 for z/OS database. It is made available internally in IBM on *as is* basis with no warranty expressed or implied

The generator is designed to be run from either TSO Batch or Windows, although the only action permitted from within Windows is RUN.

The tool can be used with all the DB2 tools:

- ▶ Change Management
Create a development and production database, make changes to development, and compare the difference.
- ▶ Performance management
Run the workload and observe with IBM Tivoli OMEGAMON XE for DB2 Performance Monitor / Expert, capture the SQL for analysis with QM (Query Monitor).
- ▶ Recovery management
Examine the logs with LAT (Log Analysis Tool), use Change Accumulation Tool to update image copies.

This tool is composed of five principal tables (GLWTDPT, GLWTEMP, GLWTPRJ, GLWTPJA and GLWTEPA), 12 supporting tables, 14 Stored Procedures (DPTADD, DPTUPD, DPTUPR, DPTMGR, DPTLCK, DPTBAL, DPTDEL, EMPADD, EMPUPD, EMPDEL, MPQRY, EMPFND, PRJADD, PRJUPD), and 12 supporting procedures, and finally, exploit views, referential integrity, partitioned tables, and triggers. See Figure A-1 on page 369.

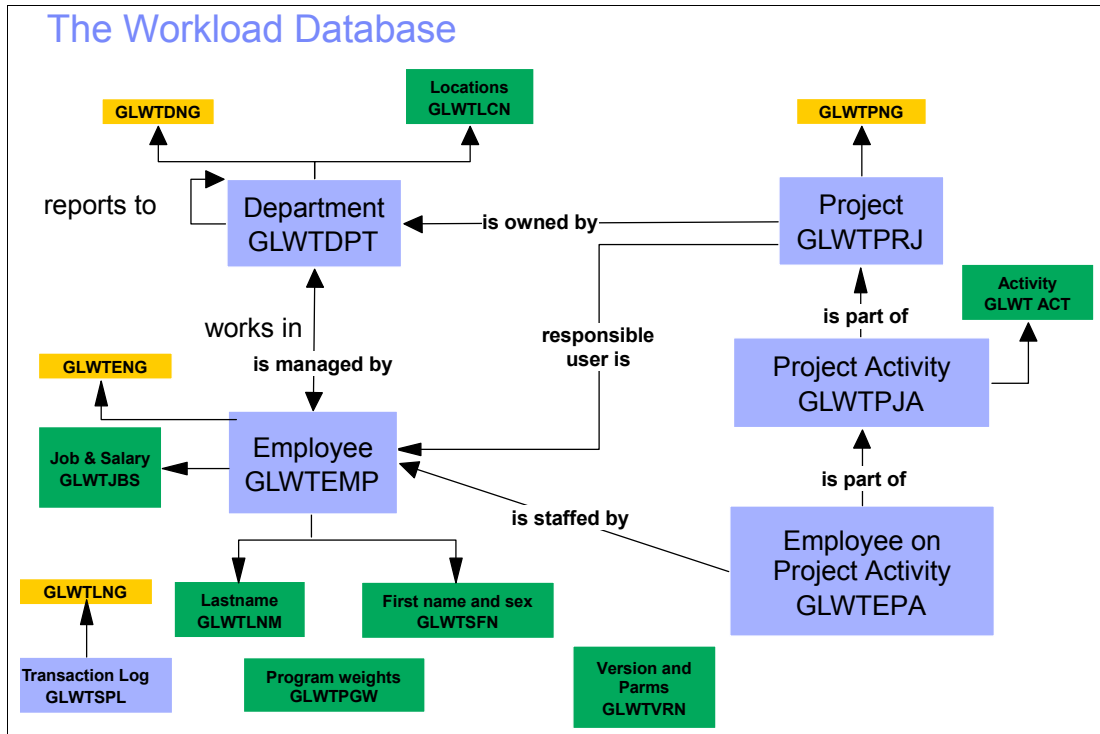


Figure A-1 The GLW database

Table A-1 summarizes the characteristics of the tables in the GLW database.

Table A-1 GLW table profiles

Table name	View name	Content	Number of columns	Number of rows (before workload)	Number of rows (after 10 minutes workload)
GLWTDPT	WLDEPT	Departments	10	0	588
GLWTEMP	WLEMP	Employees	19	0	7203
GLWTPRJ	WLPROJ	Projects	13	0	1171
GLWTPJA	WLPROJACT	Activities on each project	10	0	21078
GLWTEPA	WLEMPPROJACT	Employees assigned to each activity on each project	10	0	68184
GLWLCN	WLLOCN	Departmental locations	2	130	130
GLWJBS	WLJOBSALARY	Employee jobs and salaries	4	15	15
GLWTLNM	WLLASTNAME	Employee last names	2	351	351
GLWTSFN	WLSEXFIRSTNAME	Employee sex and first names	3	84	84
GLWACT	WLACT	Activities	3	18	18
GLWTDNG	WLDEPT_NO_GEN	Number generator for GLWTDPT	2	0	0

Table name	View name	Content	Number of columns	Number of rows (before workload)	Number of rows (after 10 minutes workload)
GLWTENG	WLEMP_NO_GEN	Number generator for GLWTEMP	2	0	0
GLWTPNG	WLPROJ_NO_GEN	Number generator for GLWTPRJ	2	0	0
GLWPNG	WLTRAN_NO_GEN	Number generator for GLWTSPL	2	0	0
GLWTSPL	WLSPLLOG	Log of each stored procedure CALL by the driver	6	0	14696
GLWTPGW	WLPROGWT	Runtime weighting for each stored procedure	2	84	84
GLWTVRN	WLVERSN	Database version control table	15	1	1

A.2.1 Getting started - Installation instructions

In this section we list the steps of our sample installation.

Prerequisite is Object REXX for Windows on your workstation.

1. Unzip GLW.V1R5.zip
2. Choose a high level qualifier (yourhlq) for the code and data.
3. Allocate the data sets listed in Table A-2 on z/OS using record format FB and record length 80 to receive the files.

Table A-2 Data set allocation on z/OS

Data set	Tracks
yourhlq.GLW.SGLWDBRM.XMIT	3
yourhlq.GLW.SGLWEXEC.XMIT	4
yourhlq.GLW.SGLWLOAD.XMIT	27
yourhlq.GLW.SGLWMLIB.XMIT	1
yourhlq.GLW.SGLWPLIB.XMIT	1
yourhlq.GLW.SGLWSAMP.XMIT	18
yourhlq.GLW.SGLWSLIB.XMIT	1
yourhlq.GLW.SGLWSRCE.XMIT	63

4. FTP the files. You may use `ftptool.bat` as shown in Example A-1.

Example: A-1 ftptool

```
C:\DOCUME~1\ADMINI~1>ftptool glw wtsc63.itso.ibm.com paolor9 fgb2001
ftp> OPEN wtsc63.itso.ibm.com
User (wtsc63.itso.ibm.com:(none)):

ftp> put GLW.SGLWDBRM.XMIT.V1R5.bin GLW.SGLWDBRM bin
ftp> put GLW.SGLWEXEC.XMIT.V1R5.bin GLW.SGLWEXEC bin
ftp> put GLW.SGLWLOAD.XMIT.V1R5.bin GLW.SGLWLOAD bin
ftp> put GLW.SGLWMLIB.XMIT.V1R5.bin GLW.SGLWMLIB bin
ftp> put GLW.SGLWPLIB.XMIT.V1R5.bin GLW.SGLWPLIB bin
ftp> put GLW.SGLWSAMP.XMIT.V1R5.bin GLW.SGLWSAMP bin
ftp> put GLW.SGLWSLIB.XMIT.V1R5.bin GLW.SGLWSLIB bin
ftp> put GLW.SGLWSRCE.XMIT.V1R5.bin GLW.SGLWSRCE bin
ftp> put GLW.SGLWSLIB.XMIT.V1R5.bin GLW.SGLWSLIB bin
ftp> put GLW.SGLWDATA.GLWLD5K.GLWDPT.TRS.v1r5.bin GLW.SGLWLD5K.GLWDPT.TRS.v1r5 bin
ftp> put GLW.SGLWDATA.GLWLD5K.GLWTEMP.TRS.v1r5.bin GLW.SGLWLD5K.GLWTEMP.TRS.v1r5 bin
ftp> put GLW.SGLWDATA.GLWLD5K.GLWTPRJ.TRS.v1r5.bin GLW.SGLWLD5K.GLWTPRJ.TRS.v1r5 bin
ftp> put GLW.SGLWDATA.GLWLD5K.GLWTPJA.TRS.v1r5.bin GLW.SGLWLD5K.GLWTPJA.TRS.v1r5 bin
ftp> put GLW.SGLWDATA.GLWLD5K.GLWTEPA.TRS.v1r5.bin GLW.SGLWLD5K.GLWTEPA.TRS.v1r5 bin
ftp> bye
C:\DOCUME~1\ADMINI~1>
```

5. Customize the JCL as shown in Example A-2 to receive the XMIT files into the PDS.

Change the `<GLWHLQ>` to `yourhlq`.

Example: A-2 Receive the XMIT files

```
//GLWRECVE JOB (99999), 'GLWRECVE', NOTIFY=&SYSUID,
//          CLASS=A, MSGCLASS=H, MSGLEVEL=(1,1), REGION=OM
//*****
//* PROPERTY OF IBM
//* (C) COPYRIGHT 2005 IBM CORP. ALL RIGHTS RESERVED.
//*
//* CHANGE THE <GLWHLQ> TO YOUR HIGH LEVEL QUALIFIER FOR THE PRODUCT
//*
//*****
//XMIT      EXEC PGM=IKJEFT01, DYNAMNBR=100
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN  DD *
RECEIVE INDS('<GLWHLQ>.GLW.SGLWDBRM.XMIT')
DSNAME('<GLWHLQ>.GLW.SGLWDBRM')
RECEIVE INDS('<GLWHLQ>.GLW.SGLWEXEC.XMIT')
DSNAME('<GLWHLQ>.GLW.SGLWEXEC')
RECEIVE INDS('<GLWHLQ>.GLW.SGLWLOAD.XMIT')
DSNAME('<GLWHLQ>.GLW.SGLWLOAD')
RECEIVE INDS('<GLWHLQ>.GLW.SGLWMLIB.XMIT')
DSNAME('<GLWHLQ>.GLW.SGLWMLIB')
RECEIVE INDS('<GLWHLQ>.GLW.SGLWPLIB.XMIT')
DSNAME('<GLWHLQ>.GLW.SGLWPLIB')
RECEIVE INDS('<GLWHLQ>.GLW.SGLWSAMP.XMIT')
DSNAME('<GLWHLQ>.GLW.SGLWSAMP')
RECEIVE INDS('<GLWHLQ>.GLW.SGLWSLIB.XMIT')
DSNAME('<GLWHLQ>.GLW.SGLWSLIB')
RECEIVE INDS('<GLWHLQ>.GLW.SGLWSRCE.XMIT')
DSNAME('<GLWHLQ>.GLW.SGLWSRCE')
```

Executing the job will create the following code data sets:

```
yourhlq.GLW.SGLWDBRM
yourhlq.GLW.SGLWEXEC
yourhlq.GLW.SGLWLOAD
yourhlq.GLW.SGLWMLIB
yourhlq.GLW.SGLWPLIB
yourhlq.GLW.SGLWSAMP
yourhlq.GLW.SGLWSLIB
yourhlq.GLW.SGLWSRCE
```

6. If you want to use the specification PRELOAD(DPT5K), you need the DB2 LOAD data, data sets GLW.SGLWLD5K.*. Ensure that the DB2 stored procedure SYSPROC.DSNUTILS is installed.

Define a set of files to receive the ftp of these data sets.

A sample job for this is yourhlq.GLW.SGLWSAMP(GLWDFTRS). See Example A-3.

You can customize this job changing GLWHLQ to yourhlq
GLW.SGLWSAMP(GLWTRSUN).

Example: A-3 Data sets allocations

```
//GLWDFTRS JOB 87152,'GLW DEFINE TRS',NOTIFY=&SYSUID,CLASS=A,
//          MSGCLASS=H,MSGLEVEL=(1,1)
//*****
/* PROPERTY OF IBM
/* (C) COPYRIGHT 2005 IBM CORP. ALL RIGHTS RESERVED.
/*
/* CHANGE THE <GLWHLQ> TO YOUR HIGH LEVEL QUALIFIER FOR THE PRODU
/*
//*****
//DELETE EXEC PGM=IEFBR14
//GLWDPT DD DSN=yourhlq.GLW.SGLWDATA.GLWLD5K.GLWDPT.TRS,
//          UNIT=SYSDA,SPACE=(TRK,0),DISP=(MOD,DELETE,DELETE)
//GLWTEMP DD DSN=yourhlq.GLW.SGLWDATA.GLWLD5K.GLWTEMP.TRS,
//          UNIT=SYSDA,SPACE=(TRK,0),DISP=(MOD,DELETE,DELETE)
//GLWTEPA DD DSN=yourhlq.GLW.SGLWDATA.GLWLD5K.GLWTEPA.TRS,
//          UNIT=SYSDA,SPACE=(TRK,0),DISP=(MOD,DELETE,DELETE)
//GLWTPJA DD DSN=yourhlq.GLW.SGLWDATA.GLWLD5K.GLWTPJA.TRS,
//          UNIT=SYSDA,SPACE=(TRK,0),DISP=(MOD,DELETE,DELETE)
//GLWTPRJ DD DSN=yourhlq.GLW.SGLWDATA.GLWLD5K.GLWTPRJ.TRS,
//          UNIT=SYSDA,SPACE=(TRK,0),DISP=(MOD,DELETE,DELETE)
/*
//DEFINE EXEC PGM=IEFBR14
//GLWDPT DD DSN=yourhlq.GLW.SGLWDATA.GLWLD5K.GLWDPT.TRS,
//          DCB=(DSORG=PS,LRECL=1024,BLKSIZE=6144,RECFM=FB),
//          UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(NEW,CATLG,DELETE)
//GLWTEMP DD DSN=yourhlq.GLW.SGLWDATA.GLWLD5K.GLWTEMP.TRS,
//          DCB=(DSORG=PS,LRECL=1024,BLKSIZE=6144,RECFM=FB),
//          UNIT=SYSDA,SPACE=(CYL,(2,1)),DISP=(NEW,CATLG,DELETE)
//GLWTEPA DD DSN=yourhlq.GLW.SGLWDATA.GLWLD5K.GLWTEPA.TRS,
//          DCB=(DSORG=PS,LRECL=1024,BLKSIZE=6144,RECFM=FB),
//          UNIT=SYSDA,SPACE=(CYL,(4,1)),DISP=(NEW,CATLG,DELETE)
//GLWTPJA DD DSN=yourhlq.GLW.SGLWDATA.GLWLD5K.GLWTPJA.TRS,
//          DCB=(DSORG=PS,LRECL=1024,BLKSIZE=6144,RECFM=FB),
//          UNIT=SYSDA,SPACE=(CYL,(4,1)),DISP=(NEW,CATLG,DELETE)
//GLWTPRJ DD DSN=yourhlq.GLW.SGLWDATA.GLWLD5K.GLWTPRJ.TRS,
//          DCB=(DSORG=PS,LRECL=1024,BLKSIZE=6144,RECFM=FB),
//          UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(NEW,CATLG,DELETE)
//
```

7. FTP the following files to the target MVS system in binary mode:

```
GLW.SGLWDATA.GLWLD5K.GLWTDPT.TRS.v1r5.bin
GLW.SGLWDATA.GLWLD5K.GLWTEMP.TRS.v1r5.bin
GLW.SGLWDATA.GLWLD5K.GLWTPRJ.TRS.v1r5.bin
GLW.SGLWDATA.GLWLD5K.GLWTPJA.TRS.v1r5.bin
GLW.SGLWDATA.GLWLD5K.GLWTEPA.TRS.v1r5.bin
```

You may use `ftptool.bat` as shown in Example A-4.

Example: A-4 ftp commands

```
ftp> put GLW.SGLWDATA.GLWLD5K.GLWTDPT.TRS.v1r5.bin 'yourhlq.GLW.SGLWLD5K.GLWTDPT.TRS' bin
ftp> put GLW.SGLWDATA.GLWLD5K.GLWTEMP.TRS.v1r5.bin 'yourhlq.GLW.SGLWLD5K.GLWTEMP.TRS' bin
ftp> put GLW.SGLWDATA.GLWLD5K.GLWTPRJ.TRS.v1r5.bin 'yourhlq.GLW.SGLWLD5K.GLWTPRJ.TRS' bin
ftp> put GLW.SGLWDATA.GLWLD5K.GLWTPJA.TRS.v1r5.bin 'yourhlq.GLW.SGLWLD5K.GLWTPJA.TRS' bin
ftp> put GLW.SGLWDATA.GLWLD5K.GLWTEPA.TRS.v1r5.bin 'yourhlq.GLW.SGLWLD5K.GLWTEPA.TRS' bin
```

8. Use TRSMMAIN to unpack to:

```
yourhlq.GLW.SGLWDATA.GLWLD5K.GLWTDPT
yourhlq.GLW.SGLWDATA.GLWLD5K.GLWTEMP
yourhlq.GLW.SGLWDATA.GLWLD5K.GLWTPRJ
yourhlq.GLW.SGLWDATA.GLWLD5K.GLWTPJA
yourhlq.GLW.SGLWDATA.GLWLD5K.GLWTEPA
```

You can customize this sample job: `yourhlq.GLW.SGLWSAMP(GLWTRSUN)`. If you do not have TRSMMAIN installed then you can download it from the following Web page:

<http://techsupport.services.ibm.com/390/>

Change `<GLWHLQ>` to `yourhlq`, submit the job, and you will get the data sets:

```
yourhlq.GLW.SGLWDATA.GLWLD5K.GLWTDPT
yourhlq.GLW.SGLWDATA.GLWLD5K.GLWTEMP
yourhlq.GLW.SGLWDATA.GLWLD5K.GLWTEPA
yourhlq.GLW.SGLWDATA.GLWLD5K.GLWTPJA
yourhlq.GLW.SGLWDATA.GLWLD5K.GLWTPRJ
```

Example A-5 shows the use of TRSMMAIN.

Example: A-5 Allocating and tersing

```
//GLWTRSUN JOB 87152,'GLW TRSMMAIN',NOTIFY=&SYSUID,CLASS=A,
//          MSGCLASS=H,MSGLEVEL=(1,1)
//*****
//* PROPERTY OF IBM
//* (C) COPYRIGHT 2005 IBM CORP. ALL RIGHTS RESERVED.
//*
//* CHANGE THE <GLWHLQ> TO YOUR HIGH LEVEL QUALIFIER FOR THE PRODUCT
//*
//*****
//DELETE EXEC PGM=IEFBR14
//GLWTDPT DD DSN=<GLWHLQ>.GLW.SGLWDATA.GLWLD5K.GLWTDPT,
//          UNIT=SYSDA,SPACE=(TRK,0),DISP=(MOD,DELETE,DELETE)
//GLWTEMP DD DSN=<GLWHLQ>.GLW.SGLWDATA.GLWLD5K.GLWTEMP,
//          UNIT=SYSDA,SPACE=(TRK,0),DISP=(MOD,DELETE,DELETE)
//GLWTEPA DD DSN=<GLWHLQ>.GLW.SGLWDATA.GLWLD5K.GLWTEPA,
//          UNIT=SYSDA,SPACE=(TRK,0),DISP=(MOD,DELETE,DELETE)
//GLWTPJA DD DSN=<GLWHLQ>.GLW.SGLWDATA.GLWLD5K.GLWTPJA,
//          UNIT=SYSDA,SPACE=(TRK,0),DISP=(MOD,DELETE,DELETE)
//GLWTPRJ DD DSN=<GLWHLQ>.GLW.SGLWDATA.GLWLD5K.GLWTPRJ,
```

```

//          UNIT=SYSDA,SPACE=(TRK,0),DISP=(MOD,DELETE,DELETE)
//*
//DEFINE   EXEC PGM=IEFBR14
//GLWTDPT DD DSN=<GLWHLQ>.GLW.SGLWDATA.GLWLD5K.GLWTDPT,
//          DCB=(DSORG=PS,LRECL=131,BLKSIZE=27998,RECFM=VB),
//          UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(NEW,CATLG,DELETE)
//GLWTEMP DD DSN=<GLWHLQ>.GLW.SGLWDATA.GLWLD5K.GLWTEMP,
//          DCB=(DSORG=PS,LRECL=180,BLKSIZE=27998,RECFM=VB),
//          UNIT=SYSDA,SPACE=(CYL,(7,1)),DISP=(NEW,CATLG,DELETE)
//GLWTEPA DD DSN=<GLWHLQ>.GLW.SGLWDATA.GLWLD5K.GLWTEPA,
//          DCB=(DSORG=PS,LRECL=111,BLKSIZE=27998,RECFM=VB),
//          UNIT=SYSDA,SPACE=(CYL,(54,1)),DISP=(NEW,CATLG,DELETE)
//GLWTPJA DD DSN=<GLWHLQ>.GLW.SGLWDATA.GLWLD5K.GLWTPJA,
//          DCB=(DSORG=PS,LRECL=129,BLKSIZE=27998,RECFM=VB),
//          UNIT=SYSDA,SPACE=(CYL,(21,1)),DISP=(NEW,CATLG,DELETE)
//GLWTPRJ DD DSN=<GLWHLQ>.GLW.SGLWDATA.GLWLD5K.GLWTPRJ,
//          DCB=(DSORG=PS,LRECL=1144,BLKSIZE=27998,RECFM=VB),
//          UNIT=SYSDA,SPACE=(CYL,(11,1)),DISP=(NEW,CATLG,DELETE)
//TERSE    PROC
//PACK     EXEC PGM=TRSMAIN,PARM='UNPACK',REGION=5M
//SYSPRINT DD SYSOUT=*
//INFILE   DD DISP=SHR,DSN=&DS..TRS
//OUTFILE  DD DISP=(OLD,CATLG),DSN=&DS
//          PEND
//*
//          EXEC TERSE,DS=<GLWHLQ>.GLW.SGLWDATA.GLWLD5K.GLWTDPT
//          EXEC TERSE,DS=<GLWHLQ>.GLW.SGLWDATA.GLWLD5K.GLWTEMP
//          EXEC TERSE,DS=<GLWHLQ>.GLW.SGLWDATA.GLWLD5K.GLWTEPA
//          EXEC TERSE,DS=<GLWHLQ>.GLW.SGLWDATA.GLWLD5K.GLWTPJA
//          EXEC TERSE,DS=<GLWHLQ>.GLW.SGLWDATA.GLWLD5K.GLWTPRJ

```

9. Setup WLM environment

Either amend an existing WLM procedure or customize the sample WLM procedure yourhlq.GLW.SGLWSAMP(WLMSAMP) to your environment. See Example A-6.

Example: A-6 WLM environment

Application Environment . . .	WLMUTIP	Required
Description	DB2 Stored Procs; DB2 Utilities	
Subsystem Type	DB2	
Procedure Name	WLMENV	
Start Parameters	DB2SSN=&IWMSSNM,NUMTCB=1,APPLENV=WLMUTIP, ,JOBNAME=&IWMSSNM.UTIP	

Example A-7 shows the JCL member for the WLM procedure.

Example: A-7 Member SYS1.PROCLIB(WLMENV)

```

//WLMENV PROC RGN=OK
//IEFPROC EXEC PGM=DSNX9WLM,REGION=&RGN,TIME=NOLIMIT,
//          PARM='&DB2SSN,&NUMTCB,&APPLENV'
// INCLUDE MEMBER=&APPLENV
//SYSPRINT DD SYSOUT=*

```

Example A-8 shows the DD for the utilities.

Example: A-8 SYS1.PROCLIB(WLMUTIP)

```
//STEPLIB DD DISP=SHR,DSN=DB9A9.SDSNEXIT
//          DD DISP=SHR,DSN=DB9A9.SDSNLOAD
//UTPRINT DD SYSOUT=*
//RNPRIN01 DD SYSOUT=*
//DSSPRINT DD SYSOUT=*
//SYSTCPT DD SYSOUT=*
//SYSIN    DD UNIT=SYSDA,SPACE=(4000,(20,20),,,ROUND)
//SYSPRINT DD UNIT=SYSDA,SPACE=(4000,(20,20),,,ROUND)
```

For details on how to setup a WLM environment, see *DB2 9 for z/OS Stored Procedures: Through the CALL and Beyond*, SG24-7604.

Note: REXX stored procedures must run in a WLM environment with NUMTCB = 1. If executed in an environment with NUMTCB>1, unpredictable results, such as an OC4 will occur.

10. Set up the DB2 buffer pools by performing one of the following tasks:

- a. Defining the buffer pools BP15 and BP16 and granting use to the appropriate group or public.

```
-ALTER BUFFERPOOL(BP15) VPSIZE(500)
-ALTER BUFFERPOOL(BP16) VPSIZE(500)
GRANT USE OF BUFFERPOOL BP15, BP16 TO PUBLIC
```

- b. Use the input parameters TSBP and IXBP to override the defaults.

```
TSBP(BP15) -
IXBP(BP16) -
```

- c. Amend the default values for TSBP and IXBP in the REXX exec

yourhlq.GLW.SGLWEXEC(GLWRUN) to use your own buffer pools and ensure that they are defined.

```
TSBP = "BP15"
IXBP = "BP16"
```

11. Set up the DB2 storage group by performing one of the following tasks:

- a. Define the group GLWG01 and granting use to the appropriate group or public.

```
CREATE STOGROUP GLWG01
  VOLUMES(SBOX3Q)
  VCAT DB9AU ;
  COMMIT;
GRANT USE OF STOGROUP GLWG01 TO PUBLIC;
```

- b. Use the input parameter STGP to override the default.

```
STGP(GLWG01)
```

- c. Amend the default value for STGP in the REXX exec

yourhlq.GLW.SGLWEXEC(GLWRUN) to use your own storage group and ensure it is defined.

```
stgp = "GLWG01"
```

12. Ensure that the TEMP database and table spaces are defined for 4 K and 8 K page sizes. The PRJADD stored procedure uses a declared temporary table and so needs the TEMP database.

13. (Optional) Define the usage tracking table GLWTLOG.

Sample DDL for a database, table space, and table can be found in the member yourhlq.GLW.SGLWSAMP(GLWD001). Amend this as required. Leave the table owner as yourhlq and the name as GLWTLOG. The purpose of this table is to gather usage information about systems such as Montpellier or Dallas where there are many users. The code checks whether the table is defined before trying to use it and so this step is optional.

14. Customize the exec yourhlq.GLW.SGLWEXEC(GLWRUN)

Add the appropriate lines for your MVS system, WLM environment names, and time-out value. See Example A-9.

Example: A-9 Customizing GLWRUN

```
glwhlqa.          = "DB2SAMP"          /* hlq */
wlmenva.SC63TS.DB9A = "WLMENV1"       /* V9 WLM environment */
timeouta.SC63TS.DB9A = 6500           /* V9 WLM environment */
glw_control_id.PA0LOR9 = 1             /* This userid has the ability to DROP
                                       databases created by others */
```

15. Customize the driver yourhlq.GLW.SGLWSAMP(GLWRUN) member to adapt to your environment.

There are four possible actions:

– BUILD

CREATE the database (DROP, if it exists)

- CREATE the stored procedures
- BIND the packages
- DEFINE the GDGs for image copies
- RUN, if runtime >0

– RUN

RUN the workload. It assumes the environment exists. This is the only valid option for remote access.

– DROP

DROP the database (if created under the same user ID)

- DROP all the stored procedures
- FREE all the packages
- DELETE all the GDGs and the image copies

– BIND

BIND the packages and RUN, if runtime>0

The description of driver program parameters are on the next three tables.

Table A-3 lists the connection options.

Table A-3 Driver program parameter - Connection

Parameter	Description	Default
DB2SSID	DB2 subsystem ID for local access	None
DB2LOCN	DB2 location for remote access	None
USERID	User ID for remote access	None
SCHEMA	The schema name. This name is used for database name, table owner name, collection name, stored procedure name, trigger schema name	GLWSAMP
LOG	SUMMARY or DDL or ALL	SUMMARY

Table A-4 lists the ACTION(BUILD) options.

Table A-4 Driver program parameter - ACTION (BUILD)

Parameter	Description	Default
RUNTIME	Set to 0 for just a build of the database and application	1
EXPLAIN	Y or N	It depends
BUILDRI	Referential integrity: DB2 - DB2 defined GLW - Application defined	DB2
PRELOAD	EMPTY: application tables start empty DPT5K: GLWDPT starts with 5000 rows and the other tables are populated accordingly	EMPTY
PERFLEV	BASE: missing IXes for poor performance TUNED: additional IXes defined	BASE
GDGLIMIT	GDG limit setting for image copies	0
COMPRESS	ASIS, Yes, No	ASIS
TRIGGER	YES, NO	YES
DATAcap	ASIS, YES, NO	ASIS
BUILDDDS	Data set holding the build definitions	hlq.GLW.SGLWSAMP
BUILDMBR	Member name of build definitions	GLWSAMP
GRANTEE	GRANT access to this user ID	PUBLIC

Table A-5 lists the ACTION(RUN) options.

Table A-5 Driver program parameter - ACTION (RUN)

Parameter	Description	Default
RUNMODE	RANDOM, FIXED	RANDOM
RUNTIME	RANDOM: runtime in minutes FIXED: repeats of RUNPROF	1
RUNPROF	It determines proportion of each transaction: STANDARD: medium growth STEADY: low growth GROWTH: high growth QUERY: query only TIMEOUT: it forces a time-out USER: modifiable by any user	STANDARD
WAITTIME	It causes a delay between each transaction	0 seconds
FLUSHBP	NONE BEFORE AFTER BOTH	BOTH

The job BUILD example is shown in Example A-10.

Example: A-10 The job BUILD example

```

//*****
//WORKLOAD EXEC PGM=IKJEFT01,DYNAMNR=100
//*****
//STEPLIB DD DSN=DB9A9.SDSNEXIT,DISP=SHR
// DD DSN=DB9A9.SDSNLOAD,DISP=SHR
//SYSEXEC DD DSN=DB2SAMP.GLW.SGLWEXEC,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
%GLWRUN DB2SSID(DB9A) -
SCHEMA(DB2GLW) -
ACTION(BUILD) -
RUNTIME(1) -
RUNMODE(FIXED) -
RUNPROF(STANDARD) -
PRELOAD(DPT5K) -
WAITTIME(0) -
FLUSHBP(NONE) -
BUILDRI(DB2) -
GDGLIMIT(1) -
EXPLAIN(N) -
GRANTEE(PUBLIC) -
PERFLEV(TUNED) -
COMPRESS(NO) -
TRIGGER(YES) -
DATACAP(NONE) -
BUILDDS(DB2SAMP.GLW.SGLWSAMP) -
DBRMLIB(DB2SAMP.GLW.SGLWSAMP) -

```

```

AUDIT(NONE) -
STORPROC(COMPC) -
RSTATLEV(BASE) -
IXCOPY(NO) -
LOG(SUMMARY) -
WLMENV(WLMENV1) -
TSBP(BP15) -
IXBP(BP16) -
STORPROC(NATIVE) -
DEBUG(0)

```

The output of this job is shown in Example A-11.

Example: A-11 Output of DB2 Workload Generator job with action BUILD

```

GLWB124I: 17:14:57 CREATE DATABASE DB2GLW BUFFERPOOL
GLWB124I: 17:14:57 CREATE TABLESPACE GLWSDPT IN
GLWB124I: 17:14:57 CREATE TABLE DB2GLW.GLWTDPT (DEPT_NO
GLWB124I: 17:14:58 CREATE TABLESPACE GLWSEMP IN
GLWB124I: 17:14:59 CREATE TABLE DB2GLW.GLWTEMP (EMP_NO
GLWB124I: 17:15:00 CREATE TABLESPACE GLWSEPA IN
...
GLWB124I: 17:15:11 CREATE PROCEDURE DB2GLW.SLEEP (IN
GLWB124I: 17:15:11 CREATE PROCEDURE DB2GLW.ACTSEL (
GLWB124I: 17:15:11 CREATE PROCEDURE DB2GLW.DPTSEL (
...

```

The job RUN example is shown in Example A-12.

Example: A-12 Job RUN example

```

//*****
//WORKLOAD EXEC PGM=IKJEFT01,DYNAMNR=100
//*****
//WORKLOAD EXEC PGM=IKJEFT01,DYNAMNR=100
//STEPLIB DD DSN=DB9A9.SDSNEXIT,DISP=SHR
// DD DSN=DB9A9.SDSNLOAD,DISP=SHR
//SYSEXEC DD DSN=DB2SAMP.GLW.SGLWEXEC,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
%GLWRUN DB2SSID(DB9A) -
SCHEMA(DB2GLW) -
ACTION(RUN) -
RUNTIME(5) -
RUNMODE(RANDOM) -
RUNPROF(STANDARD) -
PRELOAD(DPT5K) -
WAITTIME(0) -
FLUSHBP(NONE) -
BUILDRI(DB2) -
GDGLIMIT(1) -
EXPLAIN(N) -
GRANTEE(PUBLIC) -
PERFLEV(TUNED) -

```

```

COMPRESS(NO) -
TRIGGER(YES) -
DATACAP(NONE) -
BUILDDS(DB2SAMP.GLW.SGLWSAMP) -
DBRMLIB(DB2SAMP.GLW.SGLWSAMP) -
AUDIT(NONE) -
STORPROC(COMPC) -
RSTATLEV(BASE) -
IXCOPY(NO) -
LOG(SUMMARY) -
WLMENV(WLMENV1) -
TSBP(BP15) -
IXBP(BP16) -
STORPROC(NATIVE) -
DEBUG(0)

```

The output of this job is shown in Example A-13.

Example: A-13 Output of DB2 Workload Generator job with action RUN

```

GLWR141I: Execution of Stored Procedures
  1 EMPADD  17:15:24 ; Objno:  30052 ; Tran:  42603 ; SQLrc:  0
  2 PRJADD  17:15:24 ; Objno:   7511 ; Tran:  42604 ; SQLrc:  0
  3 EMPADD  17:15:24 ; Objno:  30053 ; Tran:  42605 ; SQLrc:  0
  4 DPTBAL  17:15:24 ; Objno:   5002 ; Tran:  42606 ; SQLrc:  0
...
Table row count before and after report
Table name                Before    After Difference
DB2GLW.GLWTACT             18        18         0
DB2GLW.GLWTDNG             0         0         0
DB2GLW.GLWTDPT            402       678        276
DB2GLW.GLWTEMP            5132     8643     3511
...
GLWR142I: Stored Procedure call summary
DEDLCK  did not run
DPTADD  ran    508 times at an average elapsed of    0.020 seconds
DPTBAL  ran    545 times at an average elapsed of    0.027 seconds
DPTDEL  ran    110 times at an average elapsed of    0.034 seconds
Total calls= 10354 ; Runtime=    5.0 Minutes

```

Object change management

The tool creates a database with many of the features.

Two of the tables are partitioned.

Referential integrity is defined between the five operational tables - GLWTDPT, GLWTEMP, GLWTPRJ, GLWTPJA, GLWTEPA.

You can choose to use application RI by setting the BUILDR parameter to GLW.

The reference tables and the number generator tables all use identity columns.

The number generator tables are always empty as they are used like sequences in V8, purely to generate the next number.

The employee table (GLWTEMP) has three triggers defined on it which update the department table (GLWTDPT).

You can use the TRIGGER parameter to disable the triggers.

All tables have views defined on them but note that the views cannot be used unless the user has access which is either through having SYSADM or a secondary Authid which corresponds to the owner of the view or has been granted access by one of the above.

Performance management

The workload is deliberately not tuned. By running multiple threads, either from the workstation or batch or both, you can generate time-outs and deadlocks to illustrate the ability of PE to capture and show these events. There is a stored procedure (DPTLCK) which takes a lock for 65 seconds on the GLWTDPT table and hence causes time-outs on any other thread running at the time. This program can be run by using the run profile of TIMEOUT (RUNPROF(TIMEOUT)). When using this profile it is also advisable to use the WAITTIME parameter, say 10 seconds, to let some transactions run in between the locks. The stored procedure DEDLCK can be used to create deadlocks. This procedure is invoked with the run profile of DEADLOCK. To create a deadlock, at least two threads must run concurrently on the same schema at the same time. The stored procedure EMPQRY causes RID pool failures.

The stored procedure EMPFND generates dynamic SQL.

The run parameters - RUNTIME, RUNMODE and RUNPROF together with the table GLWTPGW control how any one run behaves.

The RUNMODE farm, either RANDOM or FIXED, determines whether the sequence of stored procedures called will be repeatable or random.

This parameter also controls the behavior of the stored procedures when they need to make a choice such as selecting a location from the locations table GLWTLOC for the department table GLWTDPT.

The fixed option can be used to execute repeatable runs which have exactly the same sequence of SQL calls. This can be used for comparative measurements when changing external factors such as buffer pool sizes or object placement. Note that for this sort of trial do not run more than one thread concurrently as the predictable nature of a fixed run will be lost.

The interpretation of the run time parameter depends on the run mode.

For run mode random, the run time is the duration of the run in minutes. The number of stored procedures called will depend on how fast they execute. For run mode fixed it is a little more complex as it now determines the exact number of stored procedure calls.

This is done by summing the program weights from the GLWTPGW table for the chosen run profile, such as STANDARD, and multiplying by the run time. For example if the program weights sum to 100, and they all do except for TIMEOUT and DEADLOCK, and the run time is set to 50, then $50 * 100 = 5000$ stored procedures will be executed. A number of run profiles have been defined but you can edit the USER profile to be whatever weight combination you like.

Known issues

- ▶ On some MVS systems the "not" symbol used in the code, a \ (a backslash), is not accepted by the REXX interpreter.

The REXX interpreter uses ASCII character 170 for the logical NOT operator .

- ▶ There can be a problem with workload manager if a thread is cancelled during execution. The symptom is that a subsequent batch submission fails. If this occurs then the solution is to quiesce and resume the workload manager environment where the stored procedures were executing.



Sample configuration files for DB2 Audit Management Expert for z/OS

This appendix provides details on configuration files used in 9.3, “Installation and configuration” on page 172. For details and other files such as the Repository DDL, see the Audit Management Expert library ADH.V2R1M0.SADHSAMP.

This appendix contains the following:

- ▶ B.1, “Server configuration file” on page 384
- ▶ B.2, “Agent configuration file” on page 389
- ▶ B.3, “Audit SQL collector configuration file” on page 394

B.1 Server configuration file

A sample server configuration file showing all available server configuration parameters is included in Example B-1. The agent configuration file used for the server in this project is discussed in “Server” on page 166.

Example: B-1 DB2 Audit Management Expert for z/OS server configuration file specification

```
<!--
5655-T57
Copyright IBM Corp. 2004, 2008 All Rights Reserved.
Copyright Rocket Software, Inc. 2004 - 2008 All Rights Reserved.
-->

<!--
This is a sample configuration file for the IBM DB2 Audit
Management Expert for z/OS 2.1.0 server.

NOTE: This configuration file contains the complete set of parameters
      that can be set when configuring an ADH server. See sample
      configuration file ADHCFG5 which contains the minimum required
      parameters.

A copy of this file should be made, and customized as described below.
The job to run the server must include a DD definition referring to the
customized file. (See sample member ADHSJSRV.)

The default values for all parameters may be used.
-->

<!--
  Note: This file must have valid XML syntax.
        Extraneous characters such as sequence numbers
        will cause xml syntax errors.
-->

<server-config>

<!--
  client-listener-port must specify the IP port on which the server
  listens for connections from Audit Management Expert clients.

  Valid values are integers between 49152 and 65535, inclusive.
-->

  <client-listener-port>52522</client-listener-port>

<!--
  agent-listener-port must specify the IP port on which the server
  listens for connections from Audit Management Expert agents.

  Valid values are integers between 49152 and 65535, inclusive.
-->

  <agent-listener-port>52521</agent-listener-port>

<!--
```


log-level controls the amount of output log information that is generated by the server.

Valid values are:

0 - disable logging

S - log severe error messages only

E - log error and severe error messages

W - log warning, error, and severe error messages

I - log information, warning error, and severe error messages
(recommended)

-->

<log-level>I</log-level>

<!--

server-con-alias specifies the connection to the repository database. This is a user defined string.

-->

<server-con-alias>server</server-con-alias>

<!--

control-file-dd specifies the DD name allocated to the product control file. The name specified must be allocated in the JCL used to run the agent. The default value is DB2PARMS.

Valid values are strings conforming to DD name syntax rules.

-->

<control-file-dd>DB2PARMS</control-file-dd>

<!--

object-qualifier specifies the CREATOR of the repository tables, usually SYSTOOLS.

-->

<object-qualifier>SYSTOOLS</object-qualifier>

<!--

object-collection specifies the collection that contains the ADH packages, usually SYSTOOLS. This determines the privileges to be used to execute the SQL within Audit Management Expert.

-->

<object-collection>SYSTOOLS</object-collection>

<!--

server-repository specifies the repository location as defined on the DB2 system. This parameter must be uppercase.

-->

<server-repository>???????</server-repository>

<!--

server-usr specifies the user id for the server.

-->

<server-usr>???????</server-usr>

<!--

```

server-pwd specifies the password for the server user id.
-->

<server-pwd>???????</server-pwd>

<!--
bind-retry-max specifies the maximum number of
attempts the server should make to bind to the specified client
and agent listener ports, before exiting with an error.

Typically, there is no delay in binding. It is possible that one
or both of the ports might be in use by another application. Also,
if a previous application was using one or both of the ports and
failed, there may be a delay before the system releases the port(s)
so that they can be used by the Audit Management Expert server.

Valid values are integers greater than or equal to 0.
-->

<bind-retry-max>30</bind-retry-max>

<!--
bind-retry-delay specifies the number of seconds the
server should wait between attempts to bind to the client and
agent listener ports.

The period of time (in seconds) that the server will continue
attempts to bind is:
    bind-retry-max * bind-retry-delay

Valid values are integers greater than 0.
-->

<bind-retry-delay>10</bind-retry-delay>

<!--
The trace-* parameters cause additional tracing information to be
logged, in order to diagnose errors that may occur during product
execution.

These parameters should not be enabled unless directed by product
support, as there is a significant cost in performance when they
are enabled.

Valid values for each are true or false.
-->

<trace-network>>false</trace-network>
<trace-events>>false</trace-events>
<trace-config>>false</trace-config>

<!--
remote-repository specifies the remote server where the repository
is located. The server must be defined on the local DB2 subsystem.
It must appear in the LOCATION column of the SYSIBM.LOCATIONS
table. This parameter must be uppercase.

Note:

```

```

    You must specify this parameter if you have configured the
    Audit Management Expert repository remotely from the agent.
-->

<remote-repository>SSID</remote-repository>

<!--
    remote-repository-usr specifies the user ID on the remote system.

    Notes:
    1. You must specify this parameter if you have configured the
       Audit Management Expert repository remotely from the agent.
    2. If the remote DB2 subsystem is configured properly, this
       parameter can be omitted.
-->

<remote-repository-usr>????????</remote-repository-usr>

<!--
    remote-repository-pwd specifies the password for the user ID on
    the remote system.

    Notes:
    1. You must specify this parameter if you have configured the
       Audit Management Expert repository remotely from the agent.
    2. If the remote DB2 subsystem is configured properly, this
       parameter can be omitted.
-->

<remote-repository-pwd>????????</remote-repository-pwd>

<!--
    summarizer-refresh-interval specifies the time (in seconds)
    from the last time event records in the permanent tables are
    summarized into the Audit Management Expert report summary tables.

    Default = 1800 seconds (30 minutes)
-->
<summarizer-refresh-interval>1800</summarizer-refresh-interval>

<!--
    community-string is optional, and specifies an identifying string
    for the instance of the server. In order for agents to discover
    and connect to this server, those agents must be configured with
    the same community-string value.
-->

<community-string></community-string>

<!--
    multicast-address is optional, and specifies the UDP multicast
    address on which the server should make server announcements.

    In order for agents and clients to discover this server, they must
    be configured with the same multicast-address value.

    Valid values are IP addresses in dotted-decimal notation, in the
    range from 224.0.1.0 to 238.255.255.255, inclusive.
-->

```

```

<multicast-address>236.1.2.4</multicast-address>

<!--
multicast-port is optional, and specifies the UDP multicast
port on which the server should make server announcements.

In order for agents and clients to discover this server, they must
be configured with the same multicast-port value.

Valid values are integers between 49152 and 65535, inclusive.
-->

<multicast-port>52523</multicast-port>

<!--
multicast-interface is optional, and specifies the local network
interface address on which the server should make server
announcements. If omitted, the server makes the announcements on
all interfaces.

Valid values are IP addresses in dotted-decimal notation.
-->

<multicast-interface></multicast-interface>

<!--
multicast-ttl is optional, and specifies the UDP multicast
"time-to-live" value for the server announcements. This value
specifies the maximum number of subnets over which the announcements
will be routed. Consult your network configuration documentation
for more information.

Valid values are integers greater than 0.
-->

<multicast-ttl>5</multicast-ttl>

<!--
multicast-delay is optional, and specifies the number of seconds the
server should wait between making announcements of its presence on
the network. Smaller values result in more network traffic, but
better responsiveness to agents and clients attempting to discover
this server.

Valid values are integers greater than 0.
-->

<multicast-delay>5</multicast-delay>
<!--
description is optional and free form text which describes the
instance of the server. This description value is displayed to
users in the 'Select Server' window of the Administration UI and
can help provide additional information to allow a user to choose
between multiple available servers.
-->

```

```
<description></description>
</server-config>
```

B.2 Agent configuration file

A sample agent configuration file showing all available agent configuration parameters is included in Example B-2. The agent configuration file used for the agent in this project is discussed in “Agent” on page 173.

Example: B-2 DB2 Audit Management Expert for z/OS agent configuration file specification.

```
<!--
Licensed Materials - Property of IBM
5655-T57
Copyright IBM Corp. 2004, 2008 All Rights Reserved.
Copyright Rocket Software, Inc. 2004 - 2008 All Rights Reserved.
-->

<!--
This is a sample configuration file for the IBM DB2 Audit Management
Expert for z/OS 2.1.0 agent.

NOTE: This configuration file contains the complete set of parameters
      that can be set when configuring an ADH agent. See sample
      configuration file ADHCFGA which contains the minimum required
      parameters for an adh agent.

A copy of this file should be made, and customized as described below.
The job to run the agent must include a DD definition referring to the
customized file. (See sample member ADHSJAGT.)

The default values for all parameters may be used, except for:
    server-address,
    server-port,
    agent-monitor,
    server-repository

    which must be properly configured to enable collection of audit data
    and a connection to the Audit Management Expert server.
-->

<!--
Note: This file must have valid XML syntax.
      Extraneous characters such as sequence numbers
      will cause xml syntax errors.
-->

<agent-config>

<!--
    server-address must specify the host name or IP address (in
    dotted-decimal notation, e.g., 1.2.3.4) of the Audit Management
    Expert server to which the agent should connect.
-->
<server-address>machine.company.com</server-address>
```

```

<!--
  server-port must specify the IP port number on the Audit
  Management Expert server to which the agent should connect.

  Valid values are integers between 49152 and 65535, inclusive.
-->
<server-port>52521</server-port>

<!--
  control-file-dd specifies the DD name allocated to the product
  control file. The name specified must be allocated in the JCL used
  to run the agent. The default value is DB2PARMS.

  Valid values are strings conforming to DD name syntax rules.
-->
<control-file-dd>DB2PARMS</control-file-dd>

<!--
  log-level controls the amount of output log information that is
  generated by the agent.

  Valid values are:
  O - disable logging
  S - log severe error messages only
  E - log error and severe error messages
  W - log warning, error, and severe error messages
  I - log information, warning error, and severe error messages
  (recommended)
-->
<log-level>I</log-level>

<!--
  server-connect-retry-max specifies the maximum number of
  attempts the agent should make to connect to the server, before
  exiting with an error.

  Typically, the server should be started and available before any
  agent is started, in which case the agent will immediately connect
  on the first attempt. This parameter allows for the case when the
  server is not immediately available.

  Valid values are integers greater than or equal to 0.
-->
<server-connect-retry-max>30</server-connect-retry-max>

<!--
  server-connect-retry-delay specifies the number of seconds the
  agent should wait between attempts to connect to the server.

  The period of time (in seconds) that the agent will continue
  attempts to connect is:
  server-connect-retry-max * server-connect-retry-delay

  Valid values are integers greater than 0.
-->

```

```

<server-connect-retry-delay>10</server-connect-retry-delay>

<!--
  request-thread-timeout specifies the number of seconds a
  thread/task created to do work for a specific user should remain
  idle before exiting. Setting this value higher provides a better
  response to client requests, but consumes more resources (in the
  form of extra tasks that are not performing work).

  This value should be set high enough so that a task does not exit
  during a typical end-user client session, i.e., greater than the
  expected time between end-user actions in the client.

  Valid values are integers greater than 0.
-->
<request-thread-timeout>300</request-thread-timeout>

<!--
  uppercase-passwords specifies whether or not user IDs and
  passwords specified by end users should be folded to uppercase
  before using them to authenticate the user.

  Valid values are:
  true - uppercase user IDs and passwords
  false - use user IDs and passwords as entered by the user
-->
<uppercase-passwords>true</uppercase-passwords>

<!--
  job-poll-rate specifies the number of seconds the agent should
  wait before attempts to query the status of submitted jobs. Lower
  values provide better response time to end users, but require
  more resources on the server.

  Valid values are integers greater than 0.
-->
<job-poll-rate>5</job-poll-rate>

<!--
  agent-monitor specifies the instance name that is being monitored
  by the agent. This parameter must be uppercase.
-->
<agent-monitor>DB2</agent-monitor>

<!--
  agent-monitor-con-alias specifies the alias to be used internally
  to store/retrieve the connection.
-->
<agent-monitor-con-alias>monitor</agent-monitor-con-alias>

<!--
  agent-monitor-usr specifies the user for DB2 instance being
  monitored.
-->
<agent-monitor-usr></agent-monitor-usr>

```

```

<!--
    agent-monitor-pwd specifies the password for the DB2 instance
    being monitored.
-->
    <agent-monitor-pwd></agent-monitor-pwd>

<!--
    server-con-alias specifies the connection to the repository
    database. This is a user defined string.
-->
    <server-con-alias>server</server-con-alias>

<!--
    object-qualifier specifies the CREATOR of the repository tables,
    usually SYSTOOLS.
-->
    <object-qualifier>SYSTOOLS</object-qualifier>

<!--
    object-collection specifies the collection that contains the ADH
    packages, usually SYSTOOLS. This determines the privileges to be
    used to execute the SQL within Audit Management Expert.
-->
    <object-collection>SYSTOOLS</object-collection>

<!--
    server-repository specifies the repository location as defined on
    the DB2 system. This parameter must be uppercase.
-->
    <server-repository></server-repository>

<!--
    server-usr specifies the user id for the server.
-->
    <server-usr></server-usr>

<!--
    server-pwd specifies the password for the server user id.
-->
    <server-pwd></server-pwd>

<!--
    remote-repository specifies the remote server where the repository
    is located. The server must be defined on the local DB2 subsystem.
    It must appear in the LOCATION column of the SYSIBM.LOCATIONS
    table. This parameter must be uppercase.

```

Note:

You must specify this parameter if you have configured the Audit Management Expert repository remotely from the agent.


```

-->
<remote-repository></remote-repository>

<!--
remote-repository-usr specifies the user ID on the remote system.

Notes:
1. You must specify this parameter if you have configured the
Audit Management Expert repository remotely from the agent.
2. If the remote DB2 subsystem is configured properly, this
parameter can be omitted.
-->
<remote-repository-usr></remote-repository-usr>

<!--
remote-repository-pwd specifies the password for the user ID on
the remote system.

Notes:
1. You must specify this parameter if you have configured the
Audit Management Expert repository remotely from the agent.
2. If the remote DB2 subsystem is configured properly, this
parameter can be omitted.
-->
<remote-repository-pwd></remote-repository-pwd>

<!--
conn-ping-rate specifies the interval time between accesses
to the remote db2 to prevent timeouts during idle periods. If
a remote repository is configured, the recommended value for this
setting is 30 seconds below the remote db2 subsystem configured
IDTHTOIN ZPARM value.
The default for this setting is 90 seconds.
The ping can be disabled by setting the value to 0.
-->
<conn-ping-rate></conn-ping-rate>

<!--
multicast-address is optional, and specifies the UDP multicast
address on which the agent should listen for server announcements.
In order for the agent to discover and connect to a server, that
server must be configured with the same multicast-address value.

Server discovery is performed only if the server-address parameter
is omitted.

Valid values are IP addresses in dotted-decimal notation, in the
range from 224.0.1.0 to 238.255.255.255, inclusive.
-->

<multicast-address>236.1.2.4</multicast-address>

<!--
multicast-port is optional, and specifies the UDP multicast
port on which the agent should listen for server announcements.
In order for the agent to discover and connect to a server, that
server must be configured with the same multicast-address value.

```

Server discovery is performed only if the server-address parameter is omitted.

Valid values are integers between 49152 and 65535, inclusive.

-->

```
<multicast-port>52523</multicast-port>
```

<!--

community-string is optional, and specifies the identifying string for the instance of the server. In order for this agent to discover and connect to a server, the server's community-string must be configured with the same community-string value.

-->

```
<community-string></community-string>
```

<!--

The trace-* parameters cause additional tracing information to be logged, in order to diagnose errors that may occur during product execution.

These parameters should not be enabled unless directed by product support, as there is a significant cost in performance when they are enabled.

Valid values for each are true or false.

-->

```
<trace-csi>>false</trace-csi>
<trace-db2-attachment>>false</trace-db2-attachment>
<trace-sql>>false</trace-sql>
<trace-ifi>>false</trace-ifi>
<trace-events>>false</trace-events>
<trace-network>>false</trace-network>
<trace-config>>false</trace-config>
```

```
</agent-config>
```

B.3 Audit SQL collector configuration file

A sample Audit SQL collector configuration file showing all available configuration parameters is included in Example B-3. The agent configuration file used for the agent in this project is discussed in “Audit SQL Collector” on page 167.

Example: B-3 DB2 Audit Management Expert for z/OS agent configuration file specification

```
- - 5655-T57
- (C) COPYRIGHT ROCKET SOFTWARE, INC. 1999 - 2008 ALL RIGHTS RESERVED.
-
- DESCRIPTION: THIS IS A SAMPLE MAXIMAL ADHCFGP MEMBER
-             USED FOR AUDIT MANAGEMENT EXPERT AUDIT SQL COLLECTOR
-             STARTUP.
-             VERIFY THAT THE VALUES ON EACH PARM ARE APPROPRIATE
-             FOR YOUR ENVIRONMENT.
-
- NOTE: AFTER USING THE EDIT MACRO, VERIFY THAT NONE OF THE
```

```

-          STATEMENTS EXCEED COLUMN 72 IN LENGTH.
-
-
-
SUBSYS(#SSID)          -
AUDIT_HOSTV_DSN(#ADHASCDDLQ.AHSTV.D&LYMMDD..&INTV.) -
AUDIT_TEXT_DSN(#ADHASCDDLQ.ATEXT.D&LYMMDD..&INTV.) -
AUDIT_STATEMENT_DSN(#ADHASCDDLQ.ASTMT.D&LYMMDD..&INTV.) -
AUDIT_OBJECTS_DSN(#ADHASCDDLQ.AOBS.D&LYMMDD..&INTV.) -
AUDIT_HOSTV_SPACE_UNITS(CYLS) -
AUDIT_TEXT_SPACE_UNITS(CYLS) -
AUDIT_STATEMENT_SPACE_UNITS(CYLS) -
AUDIT_OBJECTS_SPACE_UNITS(CYLS) -
AUDIT_HOSTV_PRIMARY(05) -
AUDIT_TEXT_PRIMARY(05) -
AUDIT_STATEMENT_PRIMARY(05) -
AUDIT_OBJECTS_PRIMARY(05) -
AUDIT_HOSTV_SECONDARY(05) -
AUDIT_TEXT_SECONDARY(05) -
AUDIT_STATEMENT_SECONDARY(05) -
AUDIT_OBJECTS_SECONDARY(05) -
AUDIT_HOSTV_DATACLAS() -
AUDIT_TEXT_DATACLAS() -
AUDIT_STATEMENT_DATACLAS() -
AUDIT_OBJECTS_DATACLAS() -
AUDIT_HOSTV_MGMTCLAS() -
AUDIT_TEXT_MGMTCLAS() -
AUDIT_STATEMENT_MGMTCLAS() -
AUDIT_OBJECTS_MGMTCLAS() -
AUDIT_HOSTV_STORCLAS() -
AUDIT_TEXT_STORCLAS() -
AUDIT_STATEMENT_STORCLAS() -
AUDIT_OBJECTS_STORCLAS() -
AUDIT_HOSTV_VOLUME() -
AUDIT_TEXT_VOLUME() -
AUDIT_STATEMENT_VOLUME() -
AUDIT_OBJECTS_VOLUME()
AUDIT_HOSTV_UNITNAME() -
AUDIT_TEXT_UNITNAME() -
AUDIT_STATEMENT_UNITNAME() -
AUDIT_OBJECTS_UNITNAME() -
DEBUG(Y)              -
FORCE()               -
AUTHID(#ADHUSERID)   -
INTERVAL_MIDNIGHT(N) -
OBJECTS(Y)            -
STORCLAS()           -
MGMTCLAS()           -
DATACLAS()           -
DATASET_FULL()       -
UNITNAME()           -
VOLUME()             -
INTERVAL()

```

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see “How to get Redbooks” on page 399. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *DB2 9 for z/OS: Packages Revisited*, SG24-7688
- ▶ *DB2 9 for z/OS Performance Topics*, SG24-7473
- ▶ *DB2 9 for z/OS Stored Procedures: Through the CALL and Beyond*, SG24-7604
- ▶ *DB2 for z/OS and OS/390 : Squeezing the Most Out of Dynamic SQL*, SG24-6418
- ▶ *DB2 for z/OS: Considerations on Small and Large Packages*, REDP-4424
- ▶ *DB2 UDB for z/OS: Design Guidelines for High Performance and Availability*, SG24-7134
- ▶ *DB2 UDB for z/OS Version 8 Performance Topics*, SG24-6465
- ▶ *A Deep Blue View of DB2 Performance: IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS*, SG24-7224
- ▶ *Securing DB2 and Implementing MLS on z/OS*, SG24-6480
- ▶ *IMS V6 Security Guide*, SG24-5363
- ▶ *Designing for Solution-Based Security on z/OS*, SG24-7344
- ▶ *IBM eServer zSeries 990 (z990) Cryptography Implementation*, SG24-7070
- ▶ *IBM System z10 Enterprise Class Technical Guide*, SG24-7516
- ▶ *IBM System Storage Tape Encryption Solutions*, SG24-7320
- ▶ *IBM Virtualization Engine TS7500: Planning, Implementation, and Usage Guide*, SG24-7520
- ▶ *IMS Security Guide*, SG24-5363-00
- ▶ *z/OS Mainframe Security and Audit Management using IBM Tivoli zSecure*, SG24-7633
- ▶ *Deployment Guide Series: IBM Tivoli Security Operations Manager 4.1*, SG24-7439
- ▶ *Security on z/VM*, SG24-7471-00
- ▶ *OS/390 Security Server Audit Tool and Report Application*, SG24-4820

Other publications

These publications are also relevant as further information sources:

- ▶ *z/Architecture Principles of Operation*, SA22-7832-05
- ▶ *DB2 Version 9.1 for z/OS Administration Guide*, SC18-9840-03
- ▶ *DB2 Version 9.1 for z/OS Installation Guide*, GC18-9846-02
- ▶ *DB2 Version 9.1 for z/OS Command Reference*, SC18-9844-02
- ▶ *DB2 Version 9.1 for z/OS Application Programming and SQL Guide*, SC18-9841-01
- ▶ *DB2 Version 9.1 for z/OS Performance Monitoring and Tuning Guide*, SC18-9851-02
- ▶ *DB2 Version 9.1 for z/OS SQL Reference*, SC18-9854-04
- ▶ *DB2 Version 9.1 for z/OS XML Guide*, SC18-9858-04
- ▶ *IBM Data Encryption for IMS and DB2 Databases User's Guide Version 1 Release 1*, SC18-9549-02
- ▶ *Program Directory for IBM Data Encryption for IMS and DB2 Databases V01.1 for Use with z/OS*, GI10-8682-00
- ▶ *IBM DB2 Audit Management Expert for z/OS User's Guide Version 2 Release 1*, SC19-1302-01
- ▶ *Program Directory for IBM DB2 Audit Management Expert for z/OS V2.1 for Use with z/OS*, GI10-8771-01
- ▶ *z/OS ICSF Overview*, SA22-7519
- ▶ *z/OS ICSF System Programmer's Guide*, SA22-7520
- ▶ *z/OS ICSF Application Programmer's Guide*, SA22-7522
- ▶ *z/OS Cryptographic Services Integrated Cryptographic Service Facility Administrator's Guide*, SA22-7521-13
- ▶ *z/OS ICSF Messages*, SA22-7523
- ▶ *z/OS Trusted Key Entry Workstation User's Guide 2000*, SA22-7524
- ▶ *Planning for Multilevel Security and Common Criteria*, GA22-7509

Online resources

These Web sites are also relevant as further information sources:

- ▶ Data Governance and Compliance
http://imcomp.torolab.ibm.com/wiki/index.php/Data_Governance_and_Compliance
- ▶ The IBM Data Server Security Blueprint
<http://www-01.ibm.com/software/data/db2imstools/solutions/security-blueprint.html>
- ▶ DB2 for z/OS home page
<http://www.ibm.com/software/data/db2/zos/index.html>
- ▶ IBM Optim enterprise data management solutions
<http://www.optimsolution.com>

- ▶ Utility to allow migration from a z/OS CPACF/PCI based CKDS back to a CCF (9672/z800/z900) system
<http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS1953>
- ▶ Cattail, IBM internal file-sharing application
<http://cattail.cambridge.ibm.com/cattail/#view=collections/A3A9C9A04E523DD78E52666C7F000001>
- ▶ Developerworks - Monitoring WebSphere Applications on DB2 Servers
<http://www.ibm.com/developerworks/data/library/techarticle/0212shayer/0212shayer.html>
- ▶ The z/OS Problem Determination Upload Utility
<http://www14.software.ibm.com/webapp/set2/sas/f/zaid/pduf.html>
- ▶ IBM System Storage TS1130 Tape Drive
<http://www.ibm.com/systems/storage/tape/ts1130/index.html>

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers™, Technotes, draft publications and Additional materials, and order hardcopy Redbooks, at this Web site:

ibm.com/redbooks

Help from IBM

IBM support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Abbreviations and acronyms

AES	Advanced Encryption Standard	ICSF	Integrated Cryptographic Service Facility
AP	adjunct processor	IDS	Intrusion Detection Services
API	Application Programming Interface	IFCID	identifier called an Instrumentation Facility ID
APPN	Advanced Peer-to-Peer Networking	IFI	Instrumental Facility Interface
AS	Administrative Simplification	ITSO	International Technical Support Organization
ASC	Audit SQL Collector	JARS	Java archive files
BI	Business Intelligence	JCE	Java Cryptography Extension
BIF	built-in encryption function	KGUP	Key Generator Utility Program
BSDS	Bootstrap Data Set	LOB	line of business
CCA	Common Cryptographic Architecture	LPAR	logical partition
CCF	Cryptographic Coprocessor Feature	LTO	Linear Tape-Open
CEX2A	Cryptographic Express2 Accelerator	LUOW	logical unit of work
CEX2C	Cryptographic Express2 Coprocessor	MAC	message authentication codes
CKDS	Cryptographic Key Dataset	ME	Modulus exponent
CP	central processor	MQT	Materialized Query Tables
CPACF	Central Processor Assist for Cryptographic Functions	OLTP	online transaction processing
DBAs	database administrators	PAN	Primary Account Numbers
DBMS	database management system	PCAOB	Public Company Accounting Oversight Board
DBRC	Database Recovery Control	PCI	Payment Card Industry
DES	Data Encryption Standard	PCICA	Peripheral Component Interconnect Cryptographic Accelerator
DSMON	data security monitor	PCICC	Peripheral Component Interconnect Cryptographic Coprocessor
DSS	Data Security Standard	PCIXCC	Peripheral Component Interconnect - Extended Cryptographic Coprocessor
EBCDIC	ENCODING	PDS	partitioned data sets
EKM	Encryption Key Manager	PDSE	partitioned data sets extended
EPDM	Enterprise Performance Data Manager/MVS	PED	PIN Entry Device
ETO	Extended Terminal Option	PHI	Protected Health Information
FTP	File Transfer Protocol	PII	Personally Identifying Information
GBLA	Gramm-Leach-Bliley Act	PIPEDA	Personal Information Protection and Electronic Documents Act
GEM	Generic Event Model	PKDS	Public Key Dataset
GLBA	Gramm-Leach-Bliley Act	RACF	Resource Access Control Facility
GUI	graphical user interface	RSA	Rivest-Shamir-Adleman
HIPAA	Health Insurance Portability and Accountability Act	SEC	Securities and Exchange Commission
HMC	Hardware Management Console		
IBM	International Business Machines Corporation		

SIEM	Security information and event management
SLA	service level agreement
SMF	system management facilities
SNA	Systems Network Architecture
SOC	Security Operations Center
SSL	Secure Sockets Layer
SSM	special secure mode
SSN	Social Security numbers
TCB	task control block
TCIM	Tivoli Compliance Insight Manager
TDES	Triple Data Encryption Standard
TKE	Trusted Key Entry
TKLM	Tivoli Key Lifecycle Manager
TRUE	Task-Related User Exit
TSIEM	Tivoli Security Information and Event Manager
TSOM	Tivoli Security Operations Manager
UACC	universal access authority
UAP	user administration procedure
UR	unit of recovery
VPNs	virtual private networks
WORM	Write Once Read Many
z990	zSeries 990
zIIP	z Integrated Information Processor

Index

A

abend 0799 82
accelerator mode 130
access 7, 21, 49, 91, 102, 110, 127, 154, 164, 211, 221, 225, 272, 278, 301, 318, 354, 376
access control 8, 22, 52, 156, 291
ACF2
 monitoring 100
action 4, 24, 51, 57, 103, 117, 201, 305
adjunct processor (AP) 131
administration
 solution 100
administrative authority 60
AES (Advanced Encryption Standard) 10, 30, 129, 145, 278, 300, 316, 347
agent 62, 166, 211, 242, 269, 273
AIX 148
algorithms 39, 113, 128, 157, 316, 334
 AES 41, 334
 CEX2A and 139
 CEX2C and 130
American Encryption Standard 41
analysis
 reports 36, 211, 335
AP numbers 131
AP *See* adjunct processor
API
 security functions in CICS 102
application programs, invoking callable services 137
architecture 8–9, 106, 132, 163, 169–171, 277
archive 10, 26, 51, 105, 109, 144, 257, 274, 350
archived logs 31
assessment
 risk 6
ASYM-MK *See* asymmetric-keys master key
AT-TLS 158
audit 10, 12, 23, 34, 52, 64, 92, 100, 103, 112–113, 138, 156, 164, 211, 229, 272, 280, 321, 328, 354
 PCI 12
 policy 105, 156
 regulatory compliance 36
 report 34, 75, 107, 223, 231, 272
 reporting 16, 36, 107, 164, 222
 solution 34, 106
audit configuration 26
audit facility 26, 119
audit log 33
Audit Management Expert
 Reporting User Interface 222
audit records 26, 252
audit threats 33
auditing xx, 10, 12, 16, 22, 34, 50, 91–92, 100, 103, 138, 156, 164–165, 211, 228, 272, 354
 framework 23, 42

auditor
 tools 92
authentication 8, 12, 24, 26, 29, 49–50, 114, 132, 140, 152, 157, 291, 300, 358–359
authority level 26, 86
authorization 75, 164–165, 219
authorization ID 75, 165, 188, 224
availability xix, 3, 42, 111, 113, 293, 353

B

Basel II 147
basic 62, 152
BM Tivoli Security Information and Event Manager 105
buffer pools 60, 375
bus
 see service integration bus
business context 119
business information 127
Business Intelligence (BI) 149
business partner 143, 146
business scope 35

C

CA
 ACF2 36–37
CA *See* certificate authority
California 15
California Law 147
California Security Breach Information Act 15
callable services 81, 133, 278, 300
 CCA 135
 CCA cryptographic API 133, 136
 ISCF 137
 See also ICSF callable services
cartridge memory 145
CCA 132
CCA cryptographic API
 DES key management 134
 interoperability 133
 portability 133
 See also Common Cryptographic Architecture
CCA *See* Common Cryptographic Architecture
CCF *See* Cryptographic Coprocessor Feature
central dashboard 105
Central Processor Assist for Cryptographic Functions 128
Central Processor Assist for Cryptographic Functions (CPACF) 128
 ICSF and 136
certificate 11, 142
CEX2 129
 coprocessor mode 130
 CPACF, comparing to 132
CEX2A 128, 139, 158

- performance 130
- CEX2C 10, 128, 139, 152, 277–278, 281, 311, 316, 346, 364
 - DES key management 134
 - encryption request processing 134
 - ICSF and 136, 330
 - logical partitions and 138
 - performance data, collecting 139
- challenge
 - finding critical threats 106
- change management 11
- change tracking 103
- Chinese Remainder Theorem (CRT)
 - CEX2A 139
- CICS 37, 52–53, 102, 137, 152, 228
 - security 37, 58, 78, 102
 - using RACF commands 102
- CKDS *See* Cryptographic Key Dataset
- cleanup 53, 121
- clear keys 129, 136, 278, 301, 316, 330
 - AES-128 encryption 132
 - asymmetric encryption 132
 - RSA 131
- CMS
 - see* Central Management System
- COBOL 137
- column level 38, 60
- Common Cryptographic Architecture (CCA)
 - DES key management 134
- common cryptographic architecture (CCA) 132
 - rationale for design of 132
- common key management 133
- compliance 3, 20, 34, 50, 91, 99, 110, 143, 164, 222, 228–229, 236, 274, 321, 349
 - monitoring 12
 - posture 100
 - solution 114, 143
- components xix, 30, 50, 96, 109, 127, 151, 166, 178, 200, 211, 337, 367
- COMPRESS NO 334
- compression xix, 69, 73, 94–95, 144, 274, 311, 334
- condition 16
- confidentiality xix, 49, 51, 115–116
- configuration treats 32
- connection 24, 29, 53, 131, 157, 172, 188, 193
- console 31, 129, 290
- consolidated
 - viewing 105
- constraint 351
- CONTEXT 56, 75
- control access 52, 114, 134, 280
- coprocessors 129, 281, 283, 346
 - See also* CEX2, CEX2A, CEX2C
- correlation 77
- CP Assist for Cryptographic Functions (CPACF) 129
 - CEX2C/CEX2A, comparing to 132
- CPACF 128, 149
- CPACF *See* Central Processor Assist for Cryptographic Functions
- CREATE 75, 165, 181, 224, 226

- credentials 12, 22, 39, 51, 143
- CRM 111
- CRT *See* Chinese Remainder Theorem
- Crypto Express2 Coprocessor *See* CEX2C
- Crypto Hardware Activity report 139
- crypto instructions 136
- cryptographic coprocessor feature (CCF) 128
 - DES key management and 134
- Cryptographic Express 2 Accelerator *See* CEX2A
- Cryptographic Express 2 Coprocessor 10, 128
- cryptographic hardware 127–132, 153, 278, 300
 - comparison of 132
 - CPACF 129
 - CPAF, CEX2C, CEX2A, comparing 132
 - exploiting 138
 - monitoring 139
 - secure area within 134
 - See also* hardware cryptography
 - sysplex 140
- cryptographic key dataset (CKDS) 135
 - ICSF and 136
 - key labels and 134
- cryptographic keys *See* keys
- cryptographic operations 130, 138
- cryptographic services 133, 283
 - requests 145
- cryptographic support 152
- cryptography 10, 127, 300
 - hardware 72, 127
- CSFKEYS class 136, 138, 280, 354
- CSFSERV class 136–138, 280, 354

D

- dashboard 105
- data xix, 5, 19, 49–50, 93, 104, 110, 127, 170–171, 230, 252, 257, 274, 277–278, 299–300, 315, 345, 347
 - collection 23, 34, 83, 92, 105, 139, 165, 170, 176, 211, 271
 - transmission 8–9, 32, 131
- Data Encryption Standard (DES) 129
- Data Governance Council 6
- data key (DK) 143
- data privacy 110, 157, 300
- data security 4, 23, 85, 146, 156
- Data Security Standard (DSS) 7, 113–114, 147
- data sharing group 167, 175, 265
- data stream 51
- data threats 29
- database
 - creation 20
 - installation 156
- database system 57
- Database Tools xx
- DB2 Audit Management Expert for z/OS 92, 164, 166, 211, 223, 273
- DB2 for z/OS security themes 19
- DB2 Performance Monitor 368
- DB2 table 37, 65, 93–94, 299, 311, 335
- DB2 tools 172, 368
- DBA perspective 34

- DBADM 26, 61, 228
- decrypting 334
- defined
 - ADH#MAIN parameters 198
 - buffer pools 375
 - data server functions 49
 - EDITPROC exit 94
 - executable.1.Files 33
 - filter criteria 233, 239
 - key label 346
 - key labels 312
- delegation of administration 101
- DES algorithm 41, 130, 300
 - Crypto Hardware Activity report 140
 - keys, managing 134
- DES/TDES Enablement Feature 129
- device driver 144
- digital signature 152
- disk encryption 141
- disk space 111
- documentation 21, 37, 328, 335, 358
- domain name 56
- domains 138–139, 281
- drop 164, 177, 252, 255, 269
- DS8000 141
- DSN1COMP 335

E

- eavesdropping 159
- EDITPROC 72, 269
- elements xix, 9, 19, 92, 113, 127, 151, 292, 319, 347
- encrypting 31, 73, 135, 154, 278, 300, 320–321, 347, 349
- encryption xix, 8, 49, 93, 129, 152, 252, 278, 299, 315, 345
- encryption algorithm 32, 41, 88, 136, 154, 281, 321
- Encryption Key Manager (EKM) 144
- Encryption Tool for IMS and DB2 Databases 149
- ENDUSER 76
- ERP 111
- error 50, 116, 197, 358
 - handling 119
- event
 - archiving 350
 - collection 38, 167
 - data 10, 36, 82, 105, 113, 146, 167, 346, 350
 - type 85, 146
- executable threats 33
- external security 36
- external security manager 107
- extract 32, 115, 322

F

- FCRA/FACTA 147
- file system 269
- filter 75, 77, 92, 227, 231, 274
- FIPS 140-2 standard, Level 4 130, 138
- firewall 8–9
- foreign 12

- forensic
 - review 105
- form of attack 33–34
- FTP 10, 23, 31, 37, 202

G

- GBLA 14
- Generic Event Model (GEM) 105
- governance xix–xx, 3, 5–6, 20, 110, 113–114
- Gramm-Leach-Bliley Act (GLBA) 14, 111, 147
- granularity 55, 97, 234
- group 6, 20, 50, 53, 122, 141, 155, 167, 175, 216–217, 291, 303, 362
- group name 54

H

- hardware
 - cryptography 135
- hardware cryptographic devices 128
- hardware cryptography 128
- hash 140, 294, 357
- hashing 133
- hierarchy 60, 135
- HIPAA (Health Insurance Portability and Accountability Act) 14, 111
- historical information 113
- HTTP 103, 152

I

- IBM CCA *See* Common Cryptographic Architecture
- IBM Common Cryptographic Architecture *See* Common Cryptographic Architecture 132
- IBM Data Governance ROI calculator 45
- IBM Data Server Security Roadmap 19
- IBM Encryption Key Manager component for the Java Platform 144
- IBM System
 - z 135, 300
- IBM system
 - Storage TS1120 143
- IBM Tivoli Compliance Insight Manager
 - see* Compliance Insight Manager
- IBM Tivoli OMEGAMON XE *See* OMEGAMON XE on z/OS
- IBM Tivoli Security Operations Manager
 - see* Security Operations Manager
- IBM Tivoli zSecure
 - see* zSecure
- ICSF (Integrated Cryptographic Service Facility) xxi, 10, 69, 94, 128, 135, 138, 152, 277–278, 287, 299–300, 316, 346–347
 - audit trails 138
 - instances running 138
- ICSF administrator 94, 134, 136, 153, 349
 - keys and services, controlling 136
- ICSF callable services 135–136
 - cryptographic functions 135
- ICSF coprocessor management panel 131

- ICSF FMID 128
- ICSF panels 134
- identification 7, 22, 50, 78, 114–115, 154, 173
- IFCID 97, 183
- implicit authority 62
- IMS xix, 10, 13, 27, 37, 49, 51, 93, 113, 116, 127, 152–153, 228, 277–278, 299–300, 315, 317, 345, 347
- insider 30
- installation 11, 59–60, 152, 163, 172, 281, 292, 299, 358
 - database 156
- ICSF (Integrated Cryptographic Service Facility)
 - See also* ICSF callable services
- Integrated Cryptographic Service Facility (ICSF)
 - audit trails 138
- integrity xix, 3, 9, 34, 36, 50, 92, 111–112, 140, 157, 165, 272
- intermediary 4
- internal threat 105
- Internet 147, 157
- interoperability, CCA cryptographic API 133
- intrusion 9, 106
- IP address 56, 389
- IP network 50, 56, 157
- IPSec 158
- IPSec encryption 148

J

- Java cryptography 10, 65, 96, 137, 152
- JDBC 54–55, 113, 172
- JDBC application 76
- JVM 153

K

- KEKs *See* key-encrypting keys
- Kerberos 73
- key labels
 - DES 134
- key officer 330
- key officers 278
- key pairs 145, 278, 300
- key rotation 348
- key tokens 135, 280, 354
- keys 41, 50–51, 94, 129, 277–278, 287, 300, 316, 347–348
 - managing 70, 133, 143
 - master *See* master keys
 - sizes 41
 - types 144, 278, 288
- keystore 144–145
- keystores 142
- KGUP (Key Generator Utility Program) 278

L

- legislation 15, 127
- line of business 22
- Linear Tape-Open 143
- Linux and System z9 140
- LOADLIB 200, 269

- LOB 22
- log analysis
 - details report 265
 - summary report 264
- log analysis job
 - retrieving parameters 267
- log analysis reports
 - generating 257
 - saving 266
 - viewing 266
- log analysis reports deleting 266
- log analysis template 267
 - deleting 268
 - saving 267
- log files 24
- logical partition sharing 138
- logical partitions (LPARs) 11, 139, 164, 281
 - sharing 138
- LRSN 265
- LTO 143
- LTO Ultrium 145
- LTO4 143, 145, 148

M

- MAC 140, 296, 301
 - Crypto Hardware Activity report 140
- mainframe
 - security 11, 100, 144
- maintenance 11, 63, 135, 153, 172–173, 335, 368
- management
 - risk 13, 110, 123
- master keys 130, 288, 301
 - CEX2C 136, 138, 293
 - CEX2C card protection 130
 - DES key management and 134
 - logical partitions and 138
- message security assist (MSA)
 - instructions 129
- microcode 346
- misuse 7, 26, 113–114, 252
- model 6, 105, 110, 143
- modular exponentiation (ME) 139
- MONITOR 368
- monitoring 8, 12, 74, 77, 92, 95, 103, 164, 175, 272
 - solution 96
- MSA (message security assist) 129
- MVS 79, 83, 149, 156, 318, 354

N

- naming conventions 336
- network
 - security
 - device 142
- normalized log data 105

O

- OA20045 158
- offline 86, 112, 143, 335

online xxi, 56, 95, 113, 143, 164, 288, 363
open system 144
opening 267
operating system 9, 20, 23, 49, 81, 127, 151, 164, 278–279, 346
 support 10
operations 75, 164–165, 224
Optim Data Growth Solution 110–111
Optim Data Privacy Solution 113
Optim Database Relationship Analyzer 120
Optim solutions 109
Optim Test Data Management 111, 116
options data set 136, 139
ownership 50

P

PAN (primary account number) 38
participant 158
PARTITIONED 368
Passticket 73
Payment Card Industry (PCI) 4, 111, 114, 129, 147, 347
PCI DSS (Payment Card Industry Data Security Standard) 7
PCI X 347
PCICA *See* PCI Cryptographic Accelerator
PCI-X card 129
PCIXCC 128
PCIXCC card 72
PERFORMANCE 368
performance xix, 5, 10, 25, 37–38, 50–51, 94, 102, 110, 131, 139, 152, 170–171, 271, 281, 291, 316, 360, 368
 z/OS cryptographic workload 139
Peripheral Component Interconnect - Extended Cryptographic Coprocessor 128
Peripheral Component Interconnect - Extended Cryptographic Coprocessor (PCIXCC) 128
 DES key management 134
Peripheral Component Interconnect Cryptographic Accelerator (PCICA) 128
Peripheral Component Interconnect Cryptographic Coprocessor (PCICC) 128
 DES key management 134
permission 223, 258
PERMIT command 136, 354
PII (personally identifying information) 39
PK40178 158
PK65120 96
PK69786 335
PK75337 335
PK77147 172
PK80254 335
PK81724 335
PKA *See* public key algorithm
PKI Services 11
PL/I 137
policy
 profiles 105, 280, 354
 violation 24
portability, CCA cryptographic API 133
PQ90022 62

PQ93821 62
prepared 112, 279, 281, 317–318, 347, 360
priority 6
private keys 51, 300
privilege 29, 50, 194, 213–214, 229
problem 25, 29, 31, 55, 96, 159, 257, 354, 357
 determination 31, 96
processes 4, 9, 24, 37, 59, 92, 106, 111, 158, 293, 301, 346
providers 14, 114, 152
PTF 43941 172
public key 139, 145, 157, 300
 encryption 300
public key algorithm (PKA)
 CEX2C card support for 130
public key dataset (PKDS) 135
public keys 130, 145

Q

QUERY 368

R

RACF
 delegation 101
RACF (Resource Access Control Facility) 9–10, 30, 36, 52, 100, 135, 147, 154, 228–229, 278, 280, 311, 318, 354
 administration 11, 30, 100
 commands 79, 100, 156
 database
 copy 30
 merge 156
 unload 74, 156
 graphical user interface 101
 profile 56, 64, 136, 154, 354
 SETROPTS 137
RACF (Resource Access Control Facility) commands 136
RACLIST REFRESH 66
Rational Data Architect 21
RDEFINE command 136–137
Redbooks Web site 399
 Contact us xxi
regulatory compliance 3, 97, 105, 112, 118, 236
regulatory requirement 6
relational integrity 11
replication 29, 121, 123, 325
reporting 6, 36–37, 77, 93, 95, 103, 105, 111, 113, 154, 164, 166–167, 223, 227, 274, 324–325
 audit 37, 167, 176, 228
resource
 profile 66, 155
revoke 30, 52, 164
right sized representative test data copies 11
risk 4, 35, 43, 92, 113, 146, 164, 222, 312
 management 13, 113
Rivest-Shamir-Adleman *See* RSA
RMF *See* Resource Measurement Facility
RMF Workload Activity report 139
roles xix, 5, 97, 271–272

- row level 29, 66, 95, 334
- ROWID 93
- RSA 145
- RSA algorithm 130
 - CEX2A card and 131
- RSA keys 142, 288
- RSA operations 130
- rule 22, 30, 52, 62
- rules
 - file 33

S

- sample JCL 198, 200, 290, 317, 338
- Sarbanes 15
- Sarbanes-Oxley Act 15
- scope 35
- seclabel 68
- secret 129, 281, 300
- secret keys 140
- secure keys 72, 132, 134, 288
 - asymmetric encryption 132
 - high security environment 139
- Secure Sockets Layer 157
- security
 - auditing 12, 23, 62, 77, 91, 102, 232
 - authentication 29, 50, 142
 - authorization 29, 50, 52, 93, 291
 - CEX2 card 130
 - CEX2C and 139
 - DES key management 134
 - exposures 100, 103
 - identification 52
 - integrity 14, 146
 - J2EE 12
 - master keys 134, 292
 - real-time event information 106
 - setup 66
 - z/OS 9, 36, 49, 51, 101, 134, 166, 200, 280
- security access 291
- security administrator 37, 63, 93, 105, 154, 280, 354
- security facility 79
- security label 58, 350
- security mechanism 77, 156
- security officer (SO) 136, 138
- security operations manager 105
- security policy 8, 23, 51, 105, 156
- segregation of duties 34, 165, 272
- selection criteria 118
- sensitive data 31
 - analysis 40
- sensitive information 4, 25, 114, 159
- separation of roles 35, 272
- serial number 131
- service 11, 51, 111, 133, 152, 278, 321, 354
- service level agreement 42
- service levels 111
- SET 55, 97, 174, 181, 249, 282, 302, 323
- SETROPTS command 66, 137, 354
- setup 53, 97, 143, 182, 198
- SHA-1 130

- algorithm 130
- SHA-256 140
- sharing logical partition, individual master keys 138
- SIEM (Security Information and Event Management) 105
- signature
 - processing 158
- skills 35, 52, 100
- SLA
 - See Service Level Agreement
- SMF
 - records 12, 89, 96, 138, 156, 273, 291
- SMF records
 - ICSF audit trails 138
 - type 80 138
 - type 81 138
 - type 83 138
- SMTP 106
- sniffer 25
- SNMP 106
- SO See security officer
- Social Security numbers 38
- software requirements, System z 140
- solution
 - analysis 103
 - tasks 144
- spoofing 26
- SQLCODE 76
- SQLCODE -652 360
- SSL xix, 10, 30, 51, 129, 157
- SSL handshake 132, 158
- SSL/TLS 131, 158
- SSL/TLS protocol 130
- SSN 38
- standards
 - XML 113
- subset 22, 55, 111, 131, 274
- support 51, 79, 96, 113, 116, 140, 166, 282
- symmetric-keys master key (SYM-MK)
 - DES key management 134
 - ICSF and 136
- SYM-MK See symmetric-keys master key
- SYSADM 26, 60, 186, 194, 381
- SYSCTRL 61, 174
- sysplex, hardware cryptography 140
- system
 - integrity 120
- System SSL 157
 - software 157
- System z 8–9, 20, 41, 49, 51, 106, 127, 152, 281, 346
 - cryptography infrastructure 128
 - hardware cryptography See hardware cryptography 138
 - See also z9 server
 - software requirements 140
- System z9 131, 135, 300
 - Business Class See z9 server
 - CCA cryptographic API calls 136
 - CEX2C 140

T

table space 24, 65, 95, 257, 264, 314, 318, 350, 376
tampering 17, 26, 92
tape cartridge 143
tape drive 31, 143–145, 147–148
 encryption keys 144
 outboard encryption 147
tape encryption 143–144, 146–147
 process flow 144–145
tape library 142
TCB
 (trusted computing base) 152, 291
TCIM (Tivoli Compliance Insight Manager) 105
TCP/IP 9, 23, 56, 76, 140, 157
 encryption 144, 157
TDES (Triple-DES) 10, 30, 41, 129, 278, 281, 316, 361
 key 317
TDES algorithm 130
testing 12, 23, 111, 113, 331, 348
threat
 analysis 105
throughput 129, 158
TIMESTAMP 76
Tivoli Key Lifecycle Manager 142
Tivoli OMEGAMON XE 96, 361
Tivoli OMEGAMON XE *See* OMEGAMON XE on z/OS
Tivoli Security Operations Manager
 see Security Operations Manager
Tivoli Storage Manager 144
TKLM 142
TLS (transport layer security) 10, 30, 32, 51, 130, 158
token 50, 81, 135, 152, 280, 321, 354
 VSAM data sets 135
tokens 135
training 102, 114
transactions
 business activity 112
transport 31, 152, 301, 354
trigger 62, 95, 105, 119, 377
trusted xix, 12, 22, 34, 55, 97, 272
trusted connection 56
trusted context 22, 55
Trusted Key Entry (TKE) 134, 136
trusted user 25
TS1120 69, 143–145
TSIEM 105
TSO 368
TSOM 106

U

UAP 176, 178, 200, 212
UK41354 335
UK41355 335
UK41766 335
UK41773 335
UK44045 335
UK44642 335
Ultrium 143
unload

RACF database 156

user
 administration procedure 200, 212
 group 102, 155
 IDs
 ICSF task 135
 management 82, 100, 216
 role 29, 55, 57, 235

V

verification 12, 63, 78, 131, 294, 306, 347, 357
view 8, 42, 62, 92, 106, 113, 135, 164–165, 168, 211,
284, 308, 359, 381
virtual private networks 158
virtualization 10
virus 8, 11
VPN 158
VSAM 154, 179–180
 data set 30, 85, 135, 180, 278, 318, 346
 data set, *See also* public key dataset (PKDS)
 database 30
VSE/ESA V2.7 140
vulnerability 4, 10, 23

W

W7
 language 105
Web 7, 9, 49, 89, 139, 152
Web services 152
WebSphere 12, 24, 55, 77, 152, 325
WebSphere Application Server 30, 152
 See also SSL for WebSphere Application Server
Workload Activity report 139
Write Once Read Many (WORM) 31, 112, 147
WS-Security 152

X

XML 113, 149, 156, 173, 175, 177

Z

z/Architecture 32
z/OS xix, 9, 19, 49, 92, 103, 106, 109, 111, 127, 151,
164, 166, 211–212, 222, 252, 273, 277, 281, 300, 336,
347, 354
 audit 12, 30, 64, 74, 97, 106, 167, 257, 273
 cryptographic workload on, monitoring 139
 DES key management 134
 key storage 134
 monitoring 95
 security 9, 20, 36, 49, 51, 106, 151
z/OS V1.6 140
z/OS V1.9 148
z/OS.e V1.6/V1.7 140
z/VM
 versions 140
z/VS
 versions 140
z800/z890 platforms 135

- z9 server 130
 - CEX2 features, support for 130
 - CPs, connection to 131
- z900/z990 platforms 135
- zIIP 148
- zSecure Admin 9, 100
 - RACF
 - administration 100
 - interface 100
- zSecure Alert 103, 106
 - integration 107
 - overview 103
- zSecure Audit 9, 100, 103
 - overview 103
- zSecure CICS Toolkit
 - overview 102
- zSecure Command Verifier
 - overview 104
 - profiles 105
- zSecure Visual 101–102
 - overview 101
 - RACF
 - delegation 101



Redbooks

Securing and Auditing Data on DB2 for z/OS

(0.5" spine)
0.475" x 0.873"
250 x 459 pages



Securing and Auditing Data on DB2 for z/OS



Redbooks®

Prepare for the threat from within and without

Comply with IBM Data Server Security Blueprint

Extend the skills of data professionals

In this age of complex regulatory oversight and wide ranging threats to corporate data, securing a company's information assets from internal and external threats has become a primary focus and concern for information professionals. IBM understands these requirements and using features of the System z hardware platform, DBMS and operating elements for DB2 on z/OS, and information management tools can help to provide a defense in depth which can help to provide information confidentiality, integrity, and availability.

We start with a description of the data governance requirements, with an emphasis on IBM Data Servers Blueprint including the IBM Data Server Security Roadmap, and general elements of a complete governance approach. Next, using the elements described in the first section, we position and map the specific elements and requirements of the Blueprint based scenario to IBM portfolio of security solutions.

We then focus on some specific elements and capabilities of DB2 for z/OS and System z platform. These capabilities include elements such as network roles and trusted context, exploitation of network encryption capabilities with SSL and IPSec, and native DBMS Encryption. Included are System z hardware and z/OS operating system elements.

Having laid a solid foundation with the previous components, we then take a deeper look at two specific IBM information management tools solutions.

We build scenarios that demonstrate the use of the IBM Audit Management Expert for DB2 for z/OS. We take a deep dive look at the IBM Encryption Tool for DB2 and IMS Databases, including an exploration of the new functionality which provides coexistence with DB2 hardware assisted compression.

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