



# Agenda

- Data Security
- The Solution - Data Encryption
- Common Challenges with Data Encryption
- Data Encryption for IMS and DB2 databases
  - ◆ What protection does this offering provide?
  - ◆ How does this product fit IBM's overall strategy?
- Encryption Techniques
- How Does Data Encryption for IMS and DB2 Databases Address These Challenges?
- Prerequisite Requirements
- Offering Implementation
- What about the built-in encryption functions in DB2 for z/OS™?
- Installation and customization of Data Encryption for IMS and DB2 Databases V1.1
- A Walk-thru of Data Encryption for IMS and DB2 Databases V1.1 panels
- Summary



# Data Security

- Data security is a top issue in today's world due to:
  - Need for compliance with security legislation
    - U.S. examples
      - Health Insurance Portability and Accountability Act of 1996 (HIPAA); Health care
      - Gramm-Leach-Bliley Act of 1999 (GLBA); Financial services
  - Emergence of Storage Area Networks (SANs)
    - The need for safely storing data in a widely accessible device has increased



## Slide - Data Security (1 of 2)

Why Data Encryption for IMS and DB2 databases?

Data security is a top issue for many people. There are a couple reasons for this. One reason is that current world events have increased our need for security. Another reason is that security legislation has been enacted requiring increased security.

Here are two U.S. examples of security legislation. The first one is the Health Insurance Portability and Accountability Act (HIPAA) of 1996, which affects health care. The second, which has existed for a while, relates to the finance industry and is known as the Graham-Leach-Bliley Act of 1999.

The HIPAA Act has a deadline of April 15, 2003. This act requires health care companies to protect what is called personally identifiable information (PII).

Data Encryption for IMS and DB2 databases helps provide the security protection that the customer needs.



## Slide - Data Security (2 of 2)

Another reason why data security has become a top issue concerns storage area networks (SANs).

The model for a storage area network is one in which a pool of disk space is used by many different systems and is on the network. The network could be a company's intranet, or it could even be the internet. By having such modularity (much like grid computing) and plugging more storage into the network, a possible security exposure is presented; this is because now different systems, different applications, and different platforms are all accessing the same hardware devices that have data on them. And some of that data may be highly sensitive.



## The Solution - Data Encryption

- Data Encryption

Why an IBM solution?  
Who understands data better than IBM



## Slide - The Solution - Data Encryption

The solution for these security issues is data encryption. IBM offers the IBM Data Encryption for IMS and DB2 Databases tool.



## Common Challenges with Data Encryption

- Performance overhead
- Key management
- Application changes

Each of these challenges is addressed by the Data Encryption for IMS and DB2 Databases product and will be described later



## Slide - Common Challenges with Data Encryption

Common challenges with data encryption include the following:

- 1) Performance overhead. If I am going to encrypt and decrypt data, what is this going to cost me?
- 2) Key management. Do I need to be able to manage many different encryption keys? Who gets those keys? How does that fit in with the existing authorization scheme for my DB2 and IMS data? How is all this going to work?
- 3) Application changes. What does it mean for me to incorporate this encryption scheme within my applications. Do I need to modify all of my application code? Or do I need to redefine my objects in order to get this protection?

These are three areas that are key challenges in encrypting data. This presentation addresses how the Data Encryption for IMS and DB2 Databases tool handles these challenges.



## Data Encryption for IMS and DB2 Databases

- Provides user-customizable, pre-coded exits for encrypting IMS and DB2 data
- Exploits zSeries™ and S/390® Crypto Hardware features, which results in low overhead encryption/decryption
- Uses the ANSI Data Encryption Algorithm (DEA), also known as the U.S. National Institute of Science and Technology (NIST) Data Encryption Standard (DES) algorithm
- Works at and is customizable at the IMS segment level or DB2 table level
- Conforms to the existing OS/390® and z/OS security model



## Slide - Data Encryption for IMS and DB2 Databases

Data Encryption for IMS and DB2 Databases uses pre-coded exits to control encryption and decryption of data. On the IMS side, the IMS Segment Edit/Compression exit is used. On the DB2 side, the EDITPROC exit is used. The exits exploit z/Series and S/390 Crypto hardware features. It is unique that for the zSeries platform, hardware encryption features are built into the base processors.

The exits use the crypto hardware features. DES, the data encryption standard algorithm, is used to do the encryption. Either single DES or triple DES can be used. Triple DES is simply a higher level of protection than single DES. A higher level of protection means that it is harder to break the key for decryption.

For IMS, exits are done at the segment level. IMS exits can be different for each segment. For DB2, the exit is done at the table level.

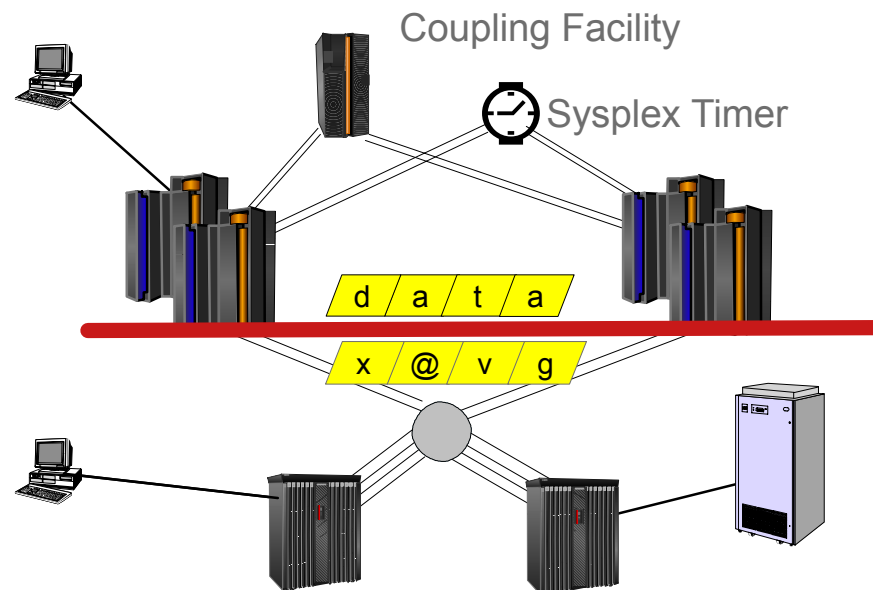
You can have different encryption keys for each table in DB2 or each segment in IMS, you can use the same encryption key for all tables or segments, or you can use one single encryption key for all the objects within your DB2 or IMS database.

This conforms to the existing OS/390 and z/OS security model.



# What protection does this offering provide?

- Data encryption on disk
  - Data on channel is encrypted (protects against channel/network sniffers)
  - Data in buffers is not encrypted
- Existing authorization controls accessing this data are unaffected
  - Assumption made that access (through the DBMS or direct access) invokes the DBMS data exits



## Slide - What protection does this offering provide

Data Encryption for IMS and DB2 Databases provides data encryption on disk.

On this slide, data above the middle line is not encrypted and data below the line is encrypted.

In this example, the encryption is not providing an additional level of security for your DB2 or IMS applications. It is providing encryption of the data on the disk.

Once the data is brought, for example, into DBMS working storage, you are using the existing DB2 and IMS authorizations to secure data. As the data is written out to disk, that is once it leaves the channel and enters the network to move to the disk, the data is encrypted.

The example on this slide shows two subsystems with disks. The circle on the bottom half of the picture might be what we have known as an ESCON director in the past. The processor on the right hand side, below the line, might also be attached to that same I/O device; however, if the processor is a 390 system that does not have the encryption key it will not be able to read the data.

## How does this product fit IBM's overall strategy?

- The on-demand operating environment has four essential characteristics:
  - ◆ Integrated
  - ◆ Open
  - ◆ Virtualized
  - ◆ Autonomic
  
- This product plays a significant role in the virtualized, or grid computing, aspect of the environment. (The virtualized environment supports a collection of computing resources to be shared and managed as if they were one large virtual computer.) In this environment:
  - ◆ Resource sharing of disks is common
  - ◆ Processors and disks are loosely coupled
  - ◆ Security of disk data can be accomplished with Data Encryption for IMS and DB2 Databases



## Slide - How does this product fit IBM's overall strategy?

IBM's overall strategy is e-business on demand. The e-business on demand environment has four characteristics, that it be integrated, open, virtualized, and autonomic.

The Data Encryption tool for IMS and DB2 Databases plays a significant role in the virtualized or grid computing.

Grid computing is defined as a virtual collection of physical resources that appear as a single, virtual, large computer.

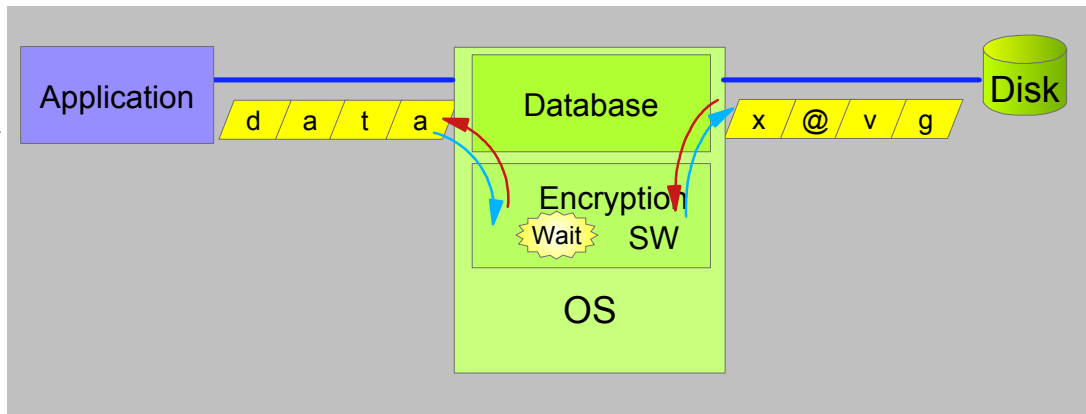
In the e-business on demand environment, resource sharing of processors, memory, and disks is essential. This is why a tool like this is very important in protecting assets between different applications and different processors by encoding the data on disk.



# Encryption Techniques

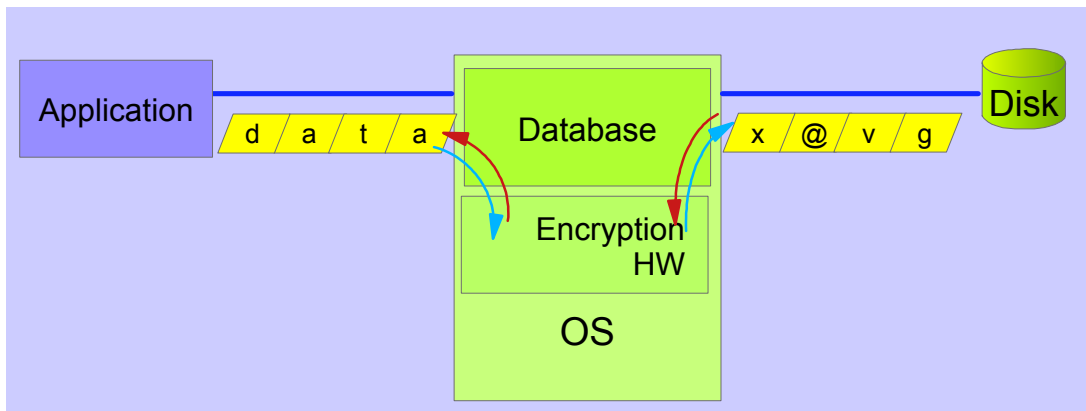
## Software

Advantage: Portability



## Hardware

Advantage: Speed



## Slide -Encryption Techniques

Now we will look at encryption itself and how encryption techniques and issues apply to the tool. In general, there are two encryption techniques, encryption using software and encryption using hardware.

If you do your encryption in software, the data on the disk is encrypted as shown in the top picture. As data comes in, it goes through some encryption software to get the data decrypted out to an application. Encryption using software is inherently slower than encryption using hardware.

The bottom picture shows the same flow, but here some type of hardware assist is used to do encryption and decryption. This is very similar to the concept of compression and decompression. We use a hardware assist to do encryption in the same way that we used a hardware assist to do compression.

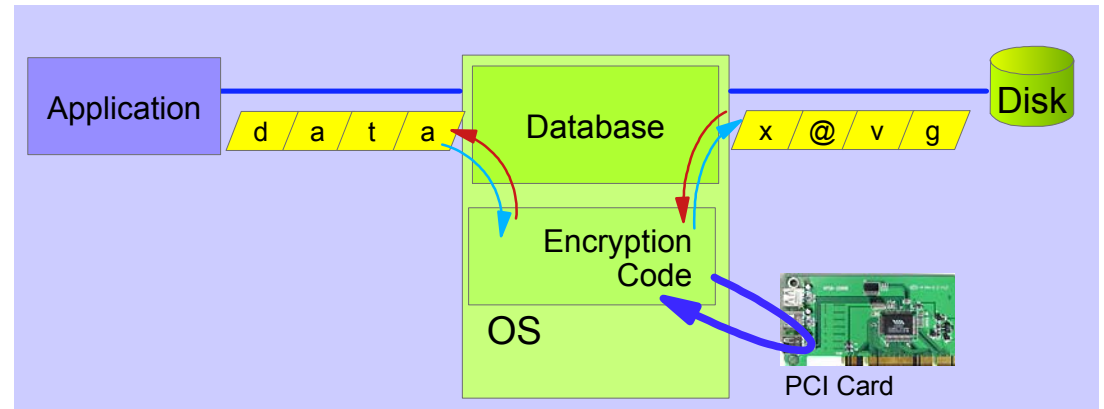
The advantage of software encryption is that you have portability. You can take the encryption software and put it on any platform as long as you have it coded in some language that lets you compile it on the different operating systems. The advantage of hardware encryption is speed.



# Hardware Encryption Techniques

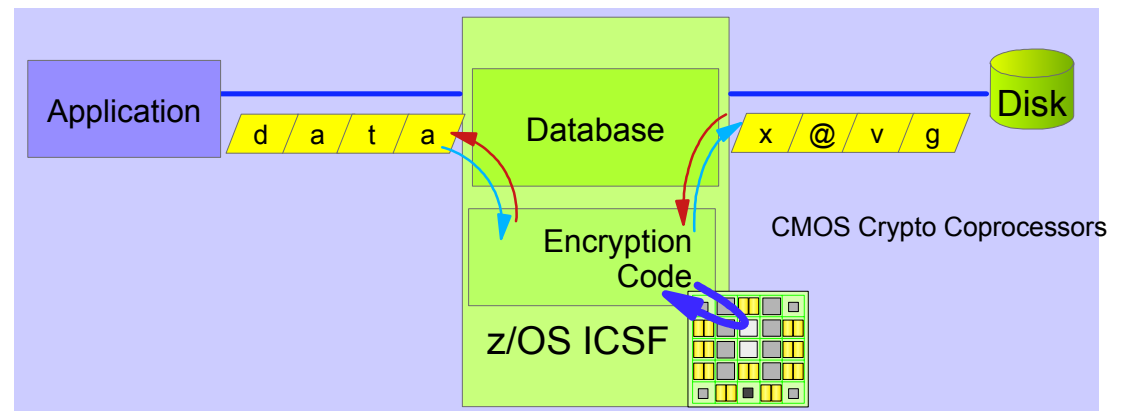
## Brand X PCI Card

Advantage: Portability



## zSeries CCF

Advantage: Speed



Crypto requests managed by z/OS (and OS/390) Integrated Cryptographic Services Facility (ISCF), which utilize on-board processors

## Slide - Hardware Encryption Techniques

There are many PCI cards available that do encryption. A PCI card is a standard hardware card that you can plug into almost any architecture. You plug in the PCI card, and you have hardware encryption. And since it is hardware encryption, you get much better speed than you would with software encryption.

One advantage that we have with the zSeries is that the cryptographic co-processor facility (CCF) gives us even more speed than a PCI card. Instead of going from a general purpose processor onto a PCI bus and into a PCI card to do the encryption and then back to the processor across the various buses, we use the crypto chips that are on the same MCM board as the rest of the general purpose processor. This is an exclusive for zSeries.

A software component in z/OS called the Integrated Cryptographic Services Facility (ICSF) is used. ICSF is invoked from the product exits, and it is ICSF that utilizes and accesses the cryptographic co-processor hardware.

With the on-board processors, we get faster speeds than we can on PCI cards.



# How Does Data Encryption for IMS and DB2 Databases Address These Challenges?

- Performance overhead
  - Lowest encryption overhead possible
  - Uses on-board processor hardware encryption
    - Infinitely faster than software encryption
    - Faster than off-board (PCI card) hardware encryption
  - Worst case laboratory measurements show 400% CPU path length increase for DB2 workloads (tablespace scan) -- this overhead is much less than the overhead for other encryption processes
  
- Key management
  - No new key management facility learning curve
  - Uses existing ICSF facility to manage encryption keys in one central repository
  
- Application changes
  - No application changes required - no passwords passed
  - System administration changes necessary only to the segment or table definition



## Slide - How Does Data Encryption for IMS and DB2 Databases Address These Challenges? (1 of 2)

The challenge for data encryption is in the areas of performance overhead, key management, and application changes. How do we address these challenges with the IBM Data Encryption for IMS and DB2 Databases tool?

**Performance overhead.** This tool has very low encryption overhead. Performance is better than any software encryption that can be performed. And it is faster than outboard or PCI-based compression; this is because of the location of the processors and because all processors share the same I/O bus. There are no calls to other pieces of hardware.

Worst case laboratory measurements show about a 400 percent CP path link increase for a DB2 workload. In this case, the workload was a table space scan. A table space scan should be the worst case performer, because every row has to be decrypted while it is being accessed. This is not the case when you are going through an index. Indexes are not encrypted in DB2. The overhead to access an index is going to be much less. In some cases, the overhead may even be unnoticeable.



## Slide - How Does Data Encryption for IMS and DB2 Databases Address These Challenges? (2 of 2)

**Key management.** There is no new key management facility to learn. Existing ICSF services are used.

**Application changes.** There are no application changes required. There are no passwords that need to be stored in applications. Passwords are passed at an exit level, and passwords are all managed.

Note, however, that in order to implement data encryption in DB2 the table must be redefined. For example, if you want to add an EDITPROC into a DB2 table you cannot alter the table and add the EDITPROC. Instead, you have to UNLOAD the data, DROP and RECREATE the table and all of its dependent objects, RELOAD the data, and REDEFINE your applications. If you have the IBM DB2 Administration Tool installed, that tool will do these steps for you (the UNLOAD, DROP and RECREATE, and RELOAD of the encrypted data by way of the exit).



# Prerequisite Requirements

- Hardware Requirements
  - Any processor capable of operating IMS Version 6 and later, and/or DB2 for OS/390 Version 6 and later
  - Any processor that supports the IBM Cryptographic Coprocessor Feature (CCF)
    - The hardware CCF modules must be enabled with configuration data (a separately orderable feature) and require a processor power-on-reset to complete the loading of the data into the crypto modules
    - Before use of the hardware encryption can occur, the hardware modules must be loaded with at least host DES master keys
  
- Software Requirements
  - IMS Version 6 or higher, and/or DB2 for OS/390 Version 6 or higher
  - OS/390 or z/OS Integrated Cryptographic Service Facility (ICSF)



## Slide - Prerequisite Requirements

Hardware requirements - any hardware that supports IMS version 6 or later or DB2 Version 6 or later is supported. This means that all the processors that DB2 V6 and IMS V6 run on have the crypto hardware. The crypto hardware must be enabled, and you must do a power-on-reset to enable the CCF modules.

Software requirements - IMS Version 6 or higher and/or DB2 Version 6 or higher. ICSF, which is a part of the z/OS operating system, is also required. ICSF is not an add-on price feature. It is a base element of z/OS and OS/390.



## Offering Implementation

- ICSF Enablement
  - Initial enablement is not trivial. It requires enablement of CCF, which is a separately-orderable hardware feature
  - One or more encryption keys can be defined and used on different segments/tables
  
- Existing objects need to be unloaded, redefined, and reloaded
  - Tools like the IBM DB2 Administration Tool can help with this process



## Slide - Offering Implementation

The initial enablement of CCF is not trivial. The CCF enablement is a separately orderable hardware feature that needs to be installed and enabled.

Existing objects need to be unloaded, redefined, and reloaded. For this, tools like the IBM DB2 Administration Tool can help.



## What about the built-in encryption functions in DB2 for z/OS?

- Different solution to a different problem
  - Column/cell based encryption
    - Application passes password in SQL statements for each cell
    - Uses the same crypto hardware (fast)
    - No password/key management provided
- Could be used in conjunction with the Data Encryption for IMS and DB2 Databases product



## Installation and customization of Data Encryption for IMS and DB2 Databases V1.1

- Easy panel-driven data entry function helps build the appropriate JCL job stream to create an encryption exit
- Implementation using the standard IMS Segment/Edit Compression and DB2 EDITPROC exits
- Before the product can be used, requires security analyst/system programmer expertise to set up an encryption key token(s) using z/OS Integrated Cryptographic Services Facility (ICSF)



## Slide - Installation and Customization of Data Encryption for IMS and DB2 Databases V1.1

The next slides describe installation and customization of the tool.

Customization is done with ease-of-use in mind. You name your DB2 or IMS system and your exits, and you enter your encryption key. The tool will then build the exits and put them into the library that you specify.

The following panels show how to give the tool the information it needs to create a job that generates the exit, assembles it, and then puts it into the exit library for you.



## Data Entry Panel Populated for an IMS Exit

```
DATA ENCRYPTION FOR IMS AND DB2 DATABAS "DOWN" is not active

Command ==> _____

Press ENTER to continue or END to exit

Specify encryption key to be implemented.
Key token . . ABCDEF

Specify encryption JCL parameters.
Jobcard . . //BILDDECX JOB (GGJ),
          . . //          'GEOF', REGION=OM, TIME=5, MSGCLASS=H, CLASS=A,
          . . //          NOTIFY=GGJ
          . . _____
          . . _____
          . . _____
          . . _____
          . . _____
CSF lib . . SYS1.LINKLIB
ZAP lib . . SYS1.MIGLIB
SMP lib . . IMSTOOL.DEC.ADECMOD0
Exit lib . . HCO.DEC.SDECLMD0
Exit name . . ENCREXIT
DBMS . . IMS (IMS or DB2)
```

### Need help?

- Press F1 if you need guidance setting up JCL parameters



# Data Entry Help Panel

```
Help panel      DATA ENCRYPTION FOR IMS AND DB2 DATABASES - BASE
COMMAND ==> _

Specify encryption key to be implemented
Key token - Enter a key token identifier that encryption will use to b
           encrypted data. Key tokens are defined by the installer of
           Integrated Cryptographic Service Facility (ICSF) .

Specify encryption JCL parameters.
Jobcard - Enter up to 5 lines of job card data.
CSF lib - The library where the Integrated Cryptographic Service Fac
          (ICSF) modules CSNBENC and CSNBDEC reside.
ZAP lib - The library where the AMASPZAP (load module zap program) r
IBM lib - The SMP library where the encryption routines have been in
Exit lib - The exit library where the encryption exit will be placed.
Exit name - The name of the encryption exit.
DBMS - Enter IMS or DB2 to indicate the database management syste
        that will use the encryption exit.
```

## More help?

- Further information is available in the User's Guide and the ICSF publications



# ISPF Edit Job Submission Panel - First Step of IMS Job

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT      GGJ.JCLOUT                      Columns 00001 00072
Command ==> _                            Scroll ==> PAGE
***** ***** Top of Data *****
000001 //BILDDECX JOB (GGJ),
000002 //          'GEOF',REGION=0M,TIME=5,MSGCLASS=H,CLASS=A,NOTIFY=GGJ
000003 //*****
000004 //* Linkedit the encryption routines into the IMS exit
000005 //*****
000006 //LINK      EXEC PGM=IEWL,PARM='LIST,XREF,RENT'
000007 //SYSPRINT DD SYSOUT=*
000008 //SYSUDUMP  DD SYSOUT=*
000009 //SDECLMDO  DD DSN=IMSTOOL.DEC.ADECMOD0,DISP=SHR
000010 //SCSFMOD0  DD DSN=SYS1.LINKLIB,DISP=SHR
000011 //SYSUT1   DD UNIT=SYSALLDA,SPACE=(1024,(50,50))
000012 //SYSLMOD   DD DSN=HCO.DEC.SDECLMDO(ENCREXIT),DISP=SHR
000013 //SYSLIN    DD *
000014  INCLUDE SDECLMDO(DECENC01)
000015  INCLUDE SCSFMOD0(CSNBENC)
000016  INCLUDE SCSFMOD0(CSNBDEC)
000017  ENTRY DECENC01
000018  NAME ENCREXIT(R)
000019 /*

```

## Note:

- This step link-edits the product-supplied IMS exit **DECENC01** into the user's exit library as a member called **ENCREXIT**

Previous task: data entry for an IMS exit

# ISPF Edit Job Submission Panel - Second Step of IMS Job

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT      GGJ.JCLOUT                      Columns 00001 00072
Command ==> _                            Scroll ==> PAGE
000020 /**
000021 //BATCHTSO EXEC PGM=IKJEFT01,DYNAMNBR=25,REGION=0M,COND=EVEN
000022 //SYSLIB DD DISP=SHR,DSN=HCO.DEC.SDECLMDO
000023 //ISPLLIB DD DISP=SHR,DSN=SYS1.MIGLIB      ** For ZAP PGM AMASPZAP **
000024 //ISPPLIB DD DISP=SHR,DSN=IMSTOOL.DEC.SDECPLIB
000025 //ISPSLIB DD DISP=SHR,DSN=IMSTOOL.DEC.SDECCLIB
000026 //ISPLLIB DD DISP=SHR,DSN=IMSTOOL.DEC.SDECLLIB
000027 //ISPTLIB DD DISP=SHR,DSN=GGJ.TEMP.ISPPROF
000028 //SYSEXEC DD DISP=SHR,DSN=IMSTOOL.DEC.SDECCEXE
000029 //ISPPROF DD DISP=SHR,DSN=GGJ.TEMP.ISPPROF
000030 //SYSTSPRT DD SYSOUT=*
000031 //ISPLOG DD SYSOUT=*,DCB=(BLKSIZE=800,LRECL=80,RECFM=FB)
000032 //SYSTSIN DD *
000033 PROFILE PREFIX(GGJ)
000034 ISPSTART CMD(%DECENC02 IMS ENCREXIT -
000035 ABCDEF )
000036 /**
***** Bottom of Data *****

```

- This step ZAPs the ICSF administrator-defined encryption key token into the IMS exit link-edited in the previous job step
- IMS applications can encrypt and decrypt data once a database has been reloaded using a new DBD defined with a COMPRTN name of **ENCREXIT**

Previous task: first step of displayed IMS job



# ISPF Edit Job Submission Panel - First Step of DB2 Job

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT      GGJ.JCLOUT                      Columns 00001 00072
Command ==>                               Scroll ==> PAGE
000001 //BILDDEX JOB (GGJ),
000002 //      'GEOF',REGION=0M,TIME=5,MSGCLASS=H,CLASS=A,NOTIFY=GGJ
000003 //*****
000004 //* Linkedit the encryption routines into the DB2 EDITPROC exit
000005 //*****
000006 //LINK      EXEC PGM=IEWL,PARM='LIST,XREF,RENT'
000007 //SYSPRINT DD SYSOUT=*
000008 //SYSUDUMP  DD SYSOUT=*
000009 //SDECLM00 DD DSN=IMSTOOL.DEC.ADECMOD0,DISP=SHR
000010 //SCSFMOD0 DD DSN=SYS1.LINKLIB,DISP=SHR
000011 //SYSUT1   DD UNIT=SYSALLDA,SPACE=(1024,(50,50))
000012 //SYSLMOD  DD DSN=HCO.DEC.SDECLM00(DB2DEXIT),DISP=SHR
000013 //SYSLIN   DD *
000014 ENTRY DECENC00
000015 INCLUDE SDECLM00(DECENC00)
000016 INCLUDE SCSFM000(CSNBENC)
000017 INCLUDE SCSFM000(CSNBDEC)
000018 NAME DB2DEXIT(R)
000019 /*
000020 /**

```

## Note:

- This step link-edits the product-supplied DB2 EDITPROC **DECENC00** into the user's exit library as a member called **DB2DEXIT**

Previous panel: data entry for a DB2 exit

## ISPF Edit Job Submission Panel - Second Step of DB2 Job

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT      GGJ.JCLOUT                      Columns 00001 00072
Command ==> _                          Scroll ==> PAGE
000020 /**
000021 //BATHTSO EXEC PGM=IKJEFT01,DYNAMNBR=25,REGION=0M,COND=EVEN
000022 //SYSLIB DD DISP=SHR,DSN=HCO.DEC.SDECLMD0
000023 //ISPLLIB DD DISP=SHR,DSN=SYS1.MIGLIB          ** For ZAP PGM AMASPZAP **
000024 //ISPPLIB DD DISP=SHR,DSN=IMSTOOL.DEC.SDECP LIB
000025 //ISPSLIB DD DISP=SHR,DSN=IMSTOOL.DEC.SDEC SLIB
000026 //ISPLIB DD DISP=SHR,DSN=IMSTOOL.DEC.SDEC MLIB
000027 //ISPTLIB DD DISP=SHR,DSN=GGJ.TEMP.ISPPROF
000028 //SYSEXEC DD DISP=SHR,DSN=IMSTOOL.DEC.SDEC CEXE
000029 //ISPPROF DD DISP=SHR,DSN=GGJ.TEMP.ISPPROF
000030 //SYSTSPRT DD SYSOUT=*
000031 //ISPLLOG DD SYSOUT=*,DCB=(BLKSIZE=800,LRECL=80,RECFM=FB)
000032 //SYSTSIN DD *
000033 PROFILE PREFIX(GGJ)
000034 ISPSTART CMD(%DECENC02 DB2 DB2DEXIT -
000035 ICSFDB2KEY )
000036 /**
***** Bottom of Data *****

```

- This step ZAPs the ICSF administrator-defined encryption key token into the IMS DB2 exit link-edited in the previous job step
- DB2 applications can now encrypt and decrypt data when they access EDITPROC **DB2DEXIT**

Previous task: first step of displayed DB2 job



# Summary

## IBM Data Encryption for IMS and DB2 Databases:

- Provides you with a single product to encrypt data for both IMS and DB2 for z/OS databases
- Lets you protect your sensitive and private data for IMS at the segment level and for DB2 at the table level

