



E13

Incremental data transfer

Wilhelm Mild

zSeries Expo

Nov. 1 - 5, 2004

Miami, FL

© IBM Corporation 2004

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and / or other counties.

CICS*	IBM*	Virtual Image
DB2*	IBM logo*	Facility
DB2 Connect	IMS	VM/ESA*
DB2 Universal	Intelligent	VSE/ESA
Database	Miner	VisualAge*
e-business logo*	Multiprise*	VTAM*
Enterprise Storage	MQSeries*	WebSphere*
Server	OS/390*	xSeries
HiperSockets	S/390*	z/Architecture
	SNAP/SHOT	z/VM
	*	zSeries

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

LINUX is a registered trademark of Linus Torvalds

Tivoli is a trademark of Tivoli Systems Inc.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries

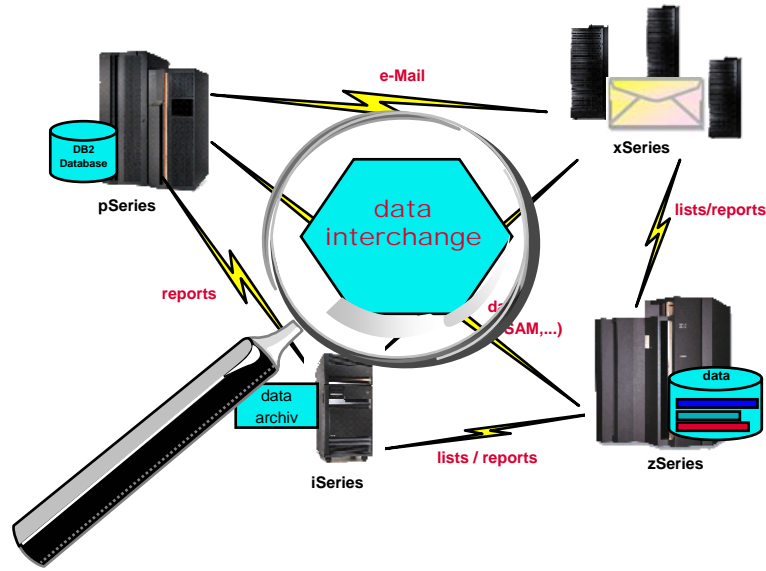
UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

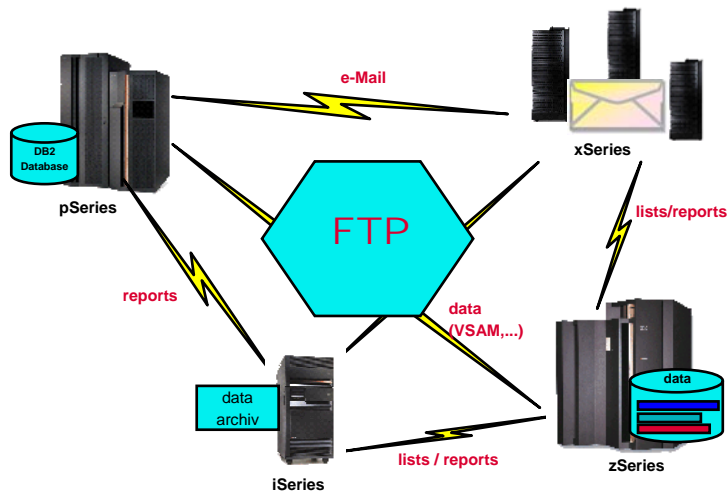
SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

Intel is a registered trademark of Intel Corporation.

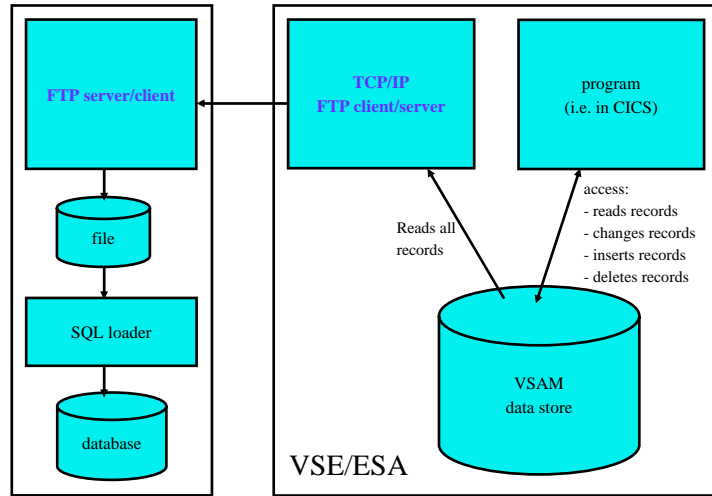
IT Environments today



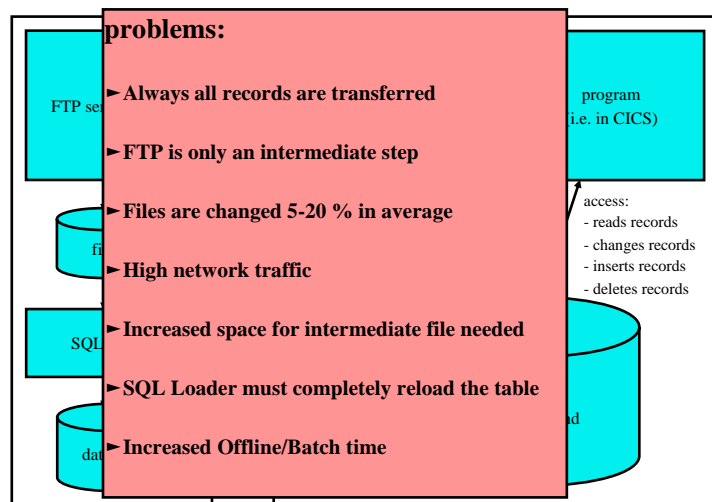
IT Environments today



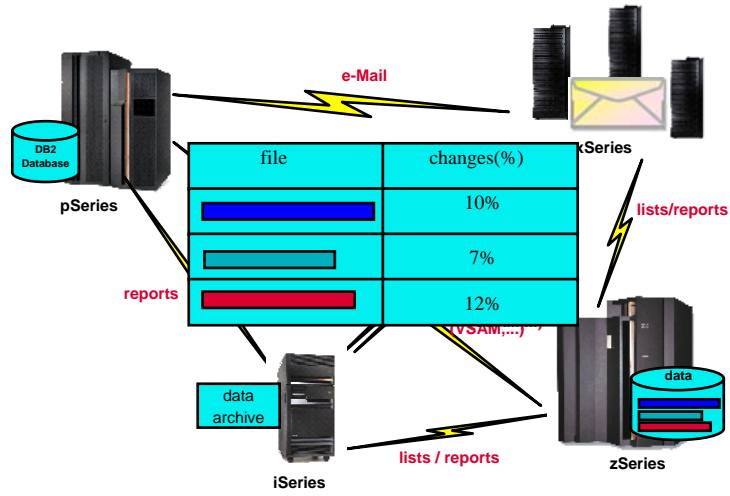
Data transfer today



Data transfer today

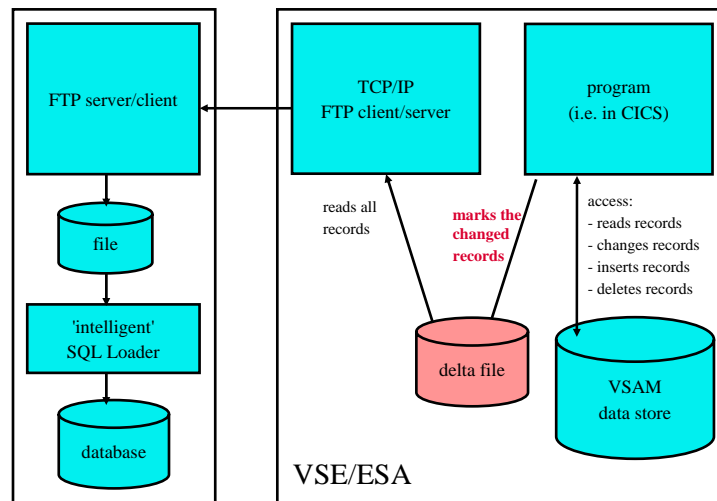


Data transfer today

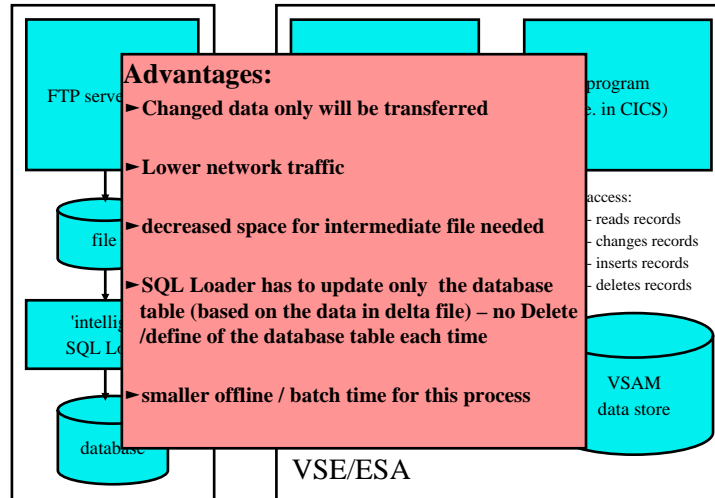


Modern solution

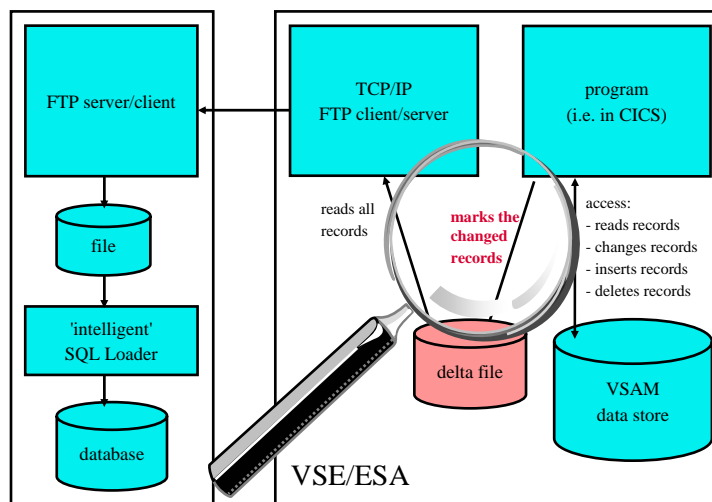
Incremental data transfer



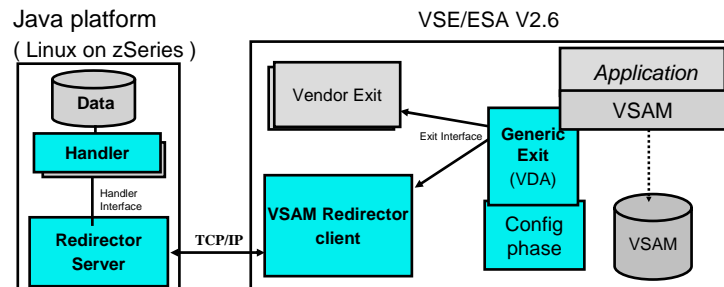
Incremental data transfer



Deeper technical look



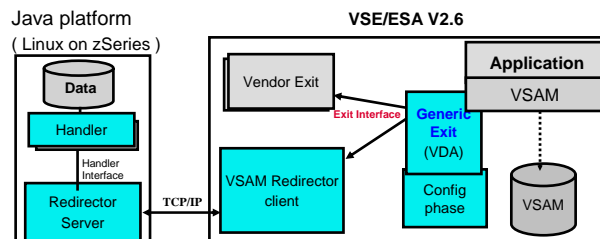
Incremental data transfer using VSE/VSAM Redirector (functional overview)



- Redirector Components:
 - Generic Exit is based on VSAM Data Access Exit (VDA)
 - Config phase - redirection properties
 - Redirector client
 - Redirector server
 - Handler

VSE/VSAM Redirector - Components

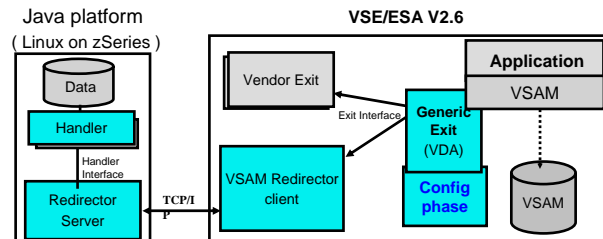
Generic Exit



- **Generic Exit**
 - based on VSAM Data Access Exit (VDA) IKQVEX01
 - VSAM requests will be intercepted (i.e. OPEN,CLOSE,GET,PUT,POINT...)
- The exit is called twice for each VSAM request.
 - before request execution
 - after request finished
- RC=0 from exit, normal VSAM processing continues
- RC=4 from exit, skip all physical VSAM access, return to caller with RC=0
- Config phase is used for decisions of further processing
- Open (documented) **Exit interface**
- To use Generic exit, no changes to VSAM programs are required.

VSE/VSAM Redirector - Components

Configuration phase



- ▶ **Config phase**
- ▶ contains parameters for the processing decision
 - ▶ cluster and catalog name
 - ▶ VSAM file should or not be redirected
 - ▶ destination specification like IP address, PORT, handler,
 - ▶ option string containing i.e. database name, userid, password
- ▶ skeleton in ICCF Library 59 (SKRD CFG) is provided
- ▶ allows integration of existing VDA Exit phases

Information in Configuration phase

- ▶ **CATALOG**= VSAM catalog name for the file to be redirected. (wildcard allowed)
(**CLUSTER**=* - required)
 - NOTE:** If the master catalog is redirected, you might not be able to startup your VSE/ESA system!
- ▶ **CATDD** = The label name of the catalog. If you enter a different value than the default (**), both the CATALOG and CATDD will be used for checking if the file is to be redirected
- ▶ **CLUSTER**= VSAM cluster name.(wildcard allowed).
- ▶ **CLUDD** = The label name of the cluster. If you enter a different value than the default (**), both the CLUSTER and CLUDD will be used for checking if the file is to be redirected
- ▶ **PART** = The partition ID (for example F4) of the partition from which redirection is only possible. The default value is PART=**.
- ▶ **NOPART** = The partition ID (for example F4) of the partition from which redirection is not possible. The default is that all partitions are available for redirection.
- ▶ **EXIT**= Name of the exit phase or 'IESREDIR'.
 - ▶ **EXIT**=phasename - the phase with this name will be invoked (integration of VDA exit)
 - ▶ **EXIT**='IESREDIR' - the redirector client will be invoked with additional parameters

Information in Configuration phase

- ▶ **OWNER**= Name of the primary VSAM data owner (VSAM or REDIRECTOR)
 - ▶ **OWNER = REDIRECTOR** - No VSAM access is done for this VSAM cluster, all requests will be redirected
 - ▶ **OWNER = VSAM**: Dual processing occurs (both VSAM processing and redirecting requests are done)

- ▶ Options *for both OWNER modes*:
- ▶ **IP**= IP address VSAM Redirector Server to handle the requests
- ▶ **PORT**= (optional, default is 2387) Port number the VSAM Redirector Server
Standard port is 2387 which is assigned by the Internet Assigned Numbers Authority (IANA).
- ▶ **HANDLER**= Name of the Java class to be started, which represents the request handler
- ▶ **OPTIONS**= A string with arbitrary, options.

- ▶ Options *for OWNER=VSAM*:
- ▶ **IGNOREERROR**= NO|YES (opt, default NO) If set to YES, no error is set if redirector server is unreachable.
- ▶ **PUTREQONLY**= NO|YES (opt, default is NO). If set to YES, only INSERT, UPDATE and DELETE requests are redirected. (useful for synchronisation purposes, excluding the requests POINT and GET).

Configuration phase

Catalog	Cluster	Exit	Owner	IP	Port	handler-name	option-string
MY.USER.CAT	MY.VSAM.FILE	IESREDIR	VSAM	10.0.0.1	4711	DB2Handler	user=xxx, pw=xxx,...
MY.USER.CAT	MY.RD.FILE	IESREDIR	REDIR	9.164.155.2	4711	DB2Handlerram	user=xxx, pw=xxx,...
VSESP.U.CAT	TEST.CLUST2	VENDOREX	n/a	n/a	n/a	n/a	n/a

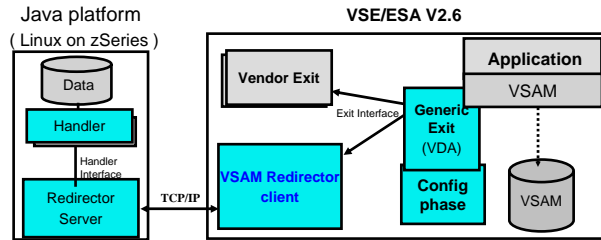
```

* $$ JOB JNM=SKRDCFG,CLASS=A,DISP=D
// JOB SKRDCFG GENERATES REDIRECTOR CONFIG PHASE
* *****
* STEP 1: ASSEMBLE AND LINK THE CONFIG TABLE *
* *****
.....
*
IESRDENT CATALOG='MY.USER.CAT',
CLUSTER='MY.VSAM.FILE',
EXIT='IESREDIR',
OWNER=VSAM,
IP='10.0.0.1',
HANDLER='com.ibm.vse.db2handler.DB2Handler',
OPTIONS='db2url=jdbc:db2:redir;db2user=hugo;
db2password=hugospw;db2table=mydata'
*
IESRDENT CATALOG='VSESP.USER.CAT',
CLUSTER='TEST.CLUSTER.',
CLUDD='DEV323'
EXIT='VENDOREX'
OPTIONS=''
*
END
/*
.....

```


VSE/VSAM Redirector - Components

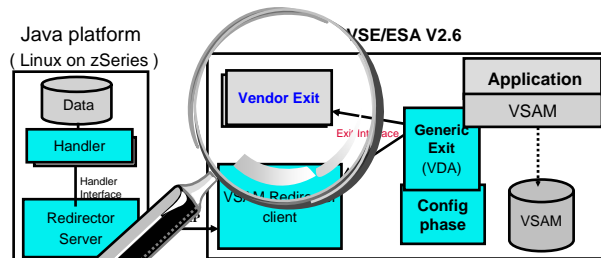
VSAM Redirector client



- ▶ **VSAM Redirector client**
 - ▶ running on VSE/ESA 2.6 in user's program partition
 - ▶ component implemented as reentrant SVA phase
 - ▶ responsible for TCP/IP connection/session handling with Redirector server

VSE/VSAM Redirector - Components

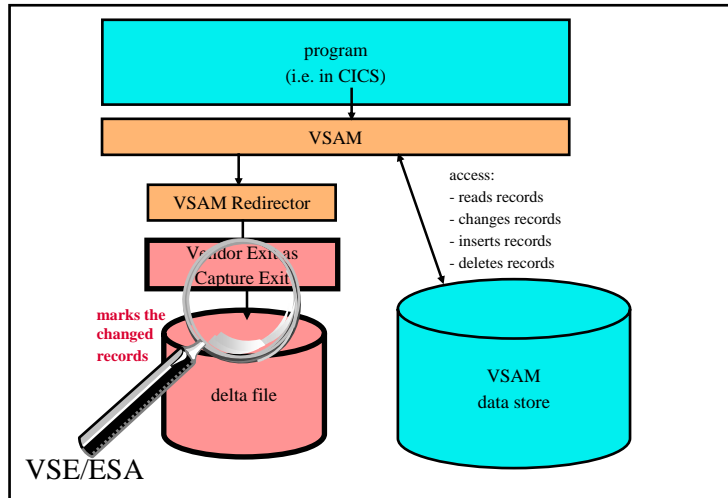
Vendor Exit



- ▶ **Vendor Exit**
 - ▶ user (vendor) written phase for data collection/transformation
 - ▶ has to comply with the documented **Exit Interface**

Note: No chaining of Vendor Exit with VSAM Redirector client supported

Deeper technical look



Journaling

or

cumulative



Rekord 1	inserted
Rekord 2	inserted
Rekord 3	inserted
Rekord 2	updated
Rekord 1	deleted
Rekord 3	updated
Rekord 4	inserted
Rekord 1	inserted
Rekord 2	updated
Rekord 4	updated
Rekord 4	deleted



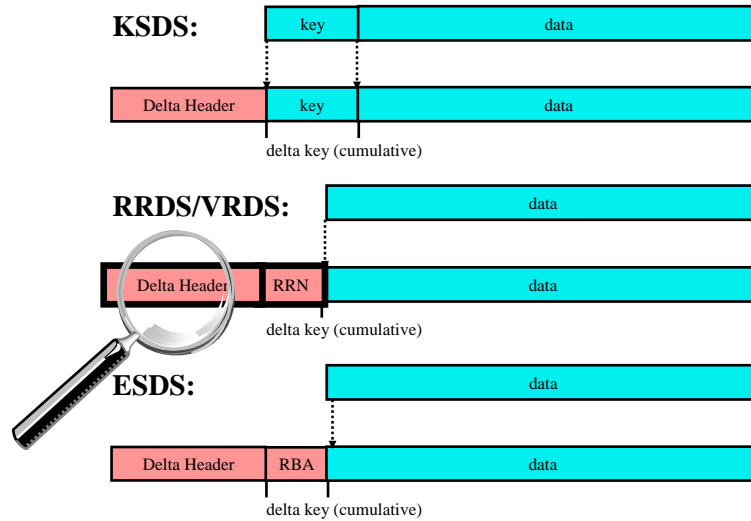
key

Rekord 1	inserted
Rekord 2	updated
Rekord 3	updated
Rekord 4	deleted

► The last version only
Of a record is stored into
the delta file



Delta Record



Delta Header (28/32 Bytes)



Offset	Parameters	Length	Description
0	TODCLOCK	8	Time of change
8	JobName	8	Job name
16	PHASEName	8	Phase name
24	PartID	2	Partition ID (i.e. F2)
26	OpCode	1	I=Insert, D=Delete, U=Update
27	Flags	1	X'01'=RRN/RBA follows
28	RRN/RBA	4	RRN/RBA (RRDS/VRDS/ESDS)

Contains information about:

- **when** changes took place (TODCLOCK)
- **who** did the changes (Job/Phase/Partition)
- **request type** of change (Insert/Delete/Update)
- **which record** was affected (key/RRN/RBA)

Why is the OpCode important ?

- The SQL Loader must know, what request was done with the delta record to make the same change in the database:
 - ▶ **Insert Rekord (OpCode=I)**
 - Insert into database
 - If already existent: execute an UPDATE
 - ▶ **Update Rekord (OpCode=U)**
 - UPDATE in database
 - If not existent: INSERT
 - ▶ **Delete Rekord (OpCode=D)**
 - DELETE in database
 - If not existent RC=0
- Depending on: journaled or cumulative process

'Intelligent' SQL Loader

- Because of the different OpCodes, an **intelligent SQL Loader** is needed
 - ▶ Has to respect the OpCodes
 - ▶ Has to treat records correspondingly:
 - insert
 - update
 - delete
 - ▶ Must be **tolerant**
 - Based on the incremental mode (journaled / cumulative)
 - i.e.: An update of a nonexistent record must be solved with an insert into the database
- Sample Java program '**VsamApply**' can help

The VsamApply sample Tool

- Java Programm 'VsamApply'
 - ▶ Executes a synchronization of a delta file with a database table
 - ▶ Needs a JDBC-driver for the database
 - ▶ The delta-file must be downloaded via FTP (BINARY) from VSE
 - ▶ The mapping information for the VSAM file must be known:
 - Which field of the VSAM record corresponds to the one in the database table
 - The field information of the delta-header can also be used (i.e. program info)
 - ▶ Can be configured via a XML file
 - ▶ Works in Initial Load Mode or Delta Mode

Additional Information

- VSE/ESA Home Page
<http://www.ibm.com/servers/eserver/zseries/os/vse/>
- VSE/ESA e-business Connectors and Utilities
<http://www-1.ibm.com/servers/eserver/zseries/os/vse/support/vseconn/conmain.htm>
- e-business Connectors User's Guide SC33-6719
<http://www-1.ibm.com/servers/eserver/zseries/os/vse/pdf/ieswue10.pdf>
- e-business Solutions for VSE/ESA SG24-5662
- e-business Connectivity for VSE/ESA SG24-5950
- CICS Transaction Server for VSE/ESA
CICS Web Support SG24-5997-00
- **NEW: Linux on zSeries: Connectors to z/OS and VSE SG24-7042 (April 2004)**



● e-business Solutions for VSE/ESA

SG24-5662



● e-business Connectivity for VSE/ESA

SG24-5950

● CICS Transaction Server for VSE/ESA
CICS Web Support

SG24-5997-00

● **NEW: Linux on zSeries: Connectors to z/OS and VSE SG24-7042 (April 2004)**

We appreciate your comments at : VSEESA@de.ibm.com