

**Using electronic mail to
automate DB2 z/OS
database copy requests**

CMG - 28 e 29 maggio 2014 - Milano, Roma

UnipolSai
ASSICURAZIONI

Agenda

- 1. UnipolSai Environment**
- 2. UnipolSai needs and problems**
- 3. The initial solution - where we started**
- 4. The new procedure - where we are**
- 5. The process in depth**
- 6. New functions - on going**



1.

UnipolSai Environment



Who is UnipolSai

New Brand UnipolSai



Division Brands



- Second Insurance Group in Italy, first in Non-Life business

- Total premiums of €15.4 billions

- About 15000 employees (Unipol gruppo)

- Company with the biggest number of branches in Italy

UnipolSai Technical Environment - Hardware

2 IBM 2827-H20 (707 + 704)

707

Production & Development (ex Unipol) :

- 7 GP + 3 zIIP + 6 zAAP + 1 ICF
- 512 GB Memory
- 1092 MSU – 8954 MIPS

704

Production (ex Fonsai) :

- 4 GP + 1 zIIP + 3 zAAP + 1 ICF
- 512 GB Memory
- 664 MSU – 5409 MIPS

14363
MIPS



1756
MSU



UnipolSai Technical Environment - Hardware

Appliance:

- IBM DB2 Analytics Accelerator for z/OS: N1001-05 (IDAA)



Storage MGM Configuration:

- DS8870 + DS8800 MM (Sync- second site)
- DS8800 + DS8700 GM (Async – third site)
- 2 X (TS7720, TS7680) Virtual Tape
- 1 X TS3500 Real Tape



Disaster recover site

IBM 2817-M15

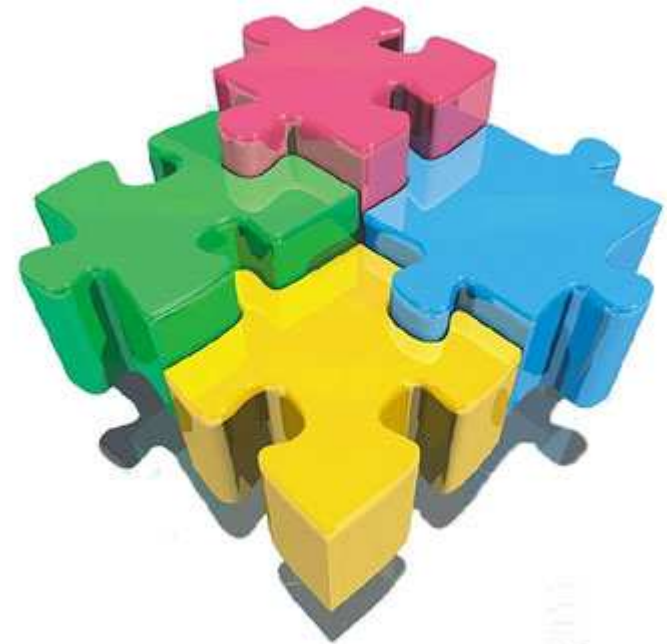
- 7 GP Processor
- 1 zIIP SE
- 5 zAAP Ses
- 1 ICF
- 165 GB Memory

Three sites configuration



UnipolSai Technical Environment - Software

- **z/OS** version 1.12
- **DB2** version 10 NFM (Data Sharing implementation on going)
10 Subsystems
- **CICS** TS 4.2
50 Subsystems
9 million transactions/daily
- **WAS** 8.5.5 + WAS 7.0.0 on z/OS
11 Application Servers (2 clustered)
6.5 million threads/daily
- WebSphere **MQ** 7.0.1



UnipolSai Application Environment

- Cobol cics/batch – static and dynamic
- Assembler
- JAVA (SQLJ/JDBC)
- DELPHI (ODBC) on Workstation
- Visual Basic - .NET on Workstation



2.

UnipolSai needs and problems



The needs: managing DB2 z/OS database copy requests

- Application Programmers frequently need to refresh DB2 data in the test environments
- Application Programmers have the authority to copy the databases they work with
- The number of daily copy requests can be huge
- Too many concurrent copies can impact the system's overall performance



The problem: how to handle the copy requests?

Therefore, we had to implement a mechanism to manage those requests

- Requests must be validated to ensure that nobody could copy unauthorized data (each Application Group can only copy DBs pertaining to them)
- Copies must be as fast and cheap as possible
- Copy jobs must run at night to minimize the impact on the online transactions



3.

**The initial
solution -
Where we started**



The initial solution

ISPF Rexx UnipolSai in-house procedure based on DSN1COPY and CROSS LOADER.

Key points:

- Check on source and target structure compatibility;
- Authorization check;
- Overnight execution of the copy requests;
- Activity Log on specific DB2 tables;
- Option to cancel the copy requests before they are executed.



The initial solution

Difficulties:

- Very hard to maintain the REXX procedure
- DB2 version upgrade impacted the REXX functionalities
- DSN1COPY: Source and Target DDLs had to be the same
- DSN1COPY: works on tablespace basis, so to copy a full database end-users had to make several requests, and that was often a problem
- CROSS LOADER: only copy on table basis
- CROSS LOADER: not a fast copy tool
- Problems whenever environmental changes were made (i.e. splitting production and test DB2 from a single LPAR to dedicated LPARS)



The point: New engine, or new car? Need a decision

- The hypothesis of amending the old ISPF procedure using **BCV5** instead of DSN1COPY was soon discarded
- The goal was to build something able to eliminate the restrictions of the old ISPF based procedure, giving development, test and QA people an easy way to obtain what they need while reducing the DBA group efforts through a high automation of the process
- Users should be able to submit their requests outside of TSO/ISPF, as a lot of requestors don't use TSO



4.

**The new
procedure - where
we are**



Decision: The new engine

To meet the requirements of high speed, low consumption, ease of use and integration in our environment we acquired the **BCV5 product**:




- Task oriented approach
- Automatic JCL creation depending on task parameters
- 6 jobs (2 on source, 4 on target), easy integration with TWS scheduler
- Compatibility checks before the copy start



New procedures related to the Copy process

With the help of the BCV5 Supplier, we decided to create a structured copy process integrating in the BCV5 skeletons two already licensed products: **Mail2ZOS** and **ZIP/390**.

The new copy procedure meet the needs:

- Easy and fast activation of the copy request process  **Mail2Zos** **1**
- Fast, linear and structured data copy process, always under control in each step  **BCV5** **2**
- Automatic reply system on the results of the copy task  **ZIP390** **3**

A new, modern car

- **Mail2ZOS** can start processes upon receipt of a formatted email message
- **ZIP/390** can send email messages
- The new procedure allows application programmers to request a database copy simply sending a formatted email
- They receive immediate notification about the acceptance or not of their request
- At the end of the overnight task, they receive email notification about the good or bad end of the copy



Advantages

- Full automation of the DB2 database copy request lifecycle
- Email messages are available 24x7 from all internet-connected devices, wherever the requestor is
- No need to remotely connect to TSO in case of urgent requests outside of the normal business hours
- Full audit of the database copy requests (who requested what, and when)



5.

**The process in
depth**



The copy request flow in depth

- Requestors send an email message to a dedicated e-mail address
- Subject must be:
COPIADB2
- Body must contain just a few parameters:
 - The name of the database to be copied
 - The subsystem name of the source environment
 - The subsystem name of the target environment



Oggetto: COPIADB2

Database DBELEPTF
Source DSH
Target DST

The copy request flow in depth

- Mail2ZOS acquires the incoming message, evaluates its syntax and, if acceptable, automatically submits the job that verifies the incoming request
- All of the needed activities to validate the incoming request are done by a REXX program included in the JCL automatically submitted by Mail2ZOS. Its main purposes are:
 - To check on the requestor's authorities;
 - To verify that no duplicate requests exist, in the same day, on the same database / subsystem (duplicate requests are denied);
 - To populate the DB2 log table containing details on each single request: Requestor, database name, source region, target region, date, result (column filled at the effective end of the copy task), copy type



The copy request flow in depth

- The BCV5 batch interface creates a new task in its own libraries
- 6 jobs are generated (Stage1 to Stage6)
- All jobs have been customized to meet our requirements, acting on the involved BCV5 skeletons
- Upon activation of the task, Stage1 and Stage2 are started sequentially:
 - Stage1 acquires the source DDLs and performs doability tests on the source objects to highlight situations which would make the copy undoable;
 - Stage2 does similar checks in the target environment to ensure that the implementation of the source data will be compatible

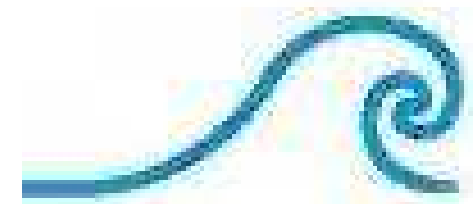


- If the checks are OK, ZIP/390 sends back an email to requestor confirming the acceptance of the request.

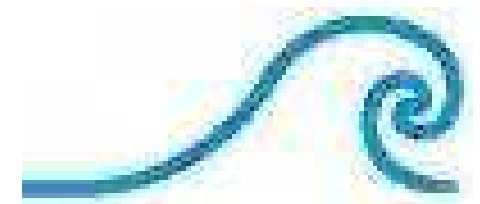
Oggetto: COPIADB2

```
RICHIESTA COPIA DATABASE DBELEFTF DSH DST  
RICHIESTA UTENTE PRESA IN CARICO !!!
```

- All jobs are moved onto the TWS libraries for overnight scheduling.



- When checks are not OK, ZIP/390 sends back an email to requestor announcing that the copy request has been rejected, attaching documentation on the reasons which led to deny the request, when available. Tasks are then removed from the BCV5 repository, so that requestors can restart from scratch, once the error has been fixed.
- The check task can issue a warning: copy can be done (BCV5 automatically switches to the LOAD utility), but requestor is advised via email so that he can fix discrepancies between source and target, wherever possible.



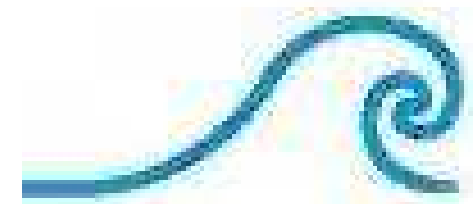
The copy request flow in depth

2 - 3

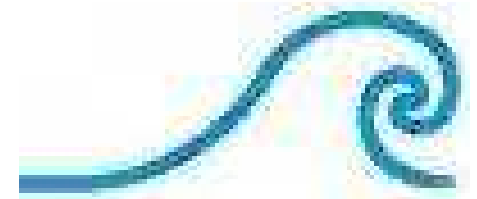
- All of the accepted tasks are executed overnight by TWS (always restarting from Stage1, to re-execute all checks on the compatibility between source and target).
- Email notifications are sent by ZIP/390 to requestor about the good or bad end of the copy.

Oggetto: COPIADB2 DSH/DST/YHTELEP6/DBELEPTF

```
COPIA DBELEPTF SENZA STRUTTURE  
Copia eseguita correttamente
```



- In case of failure, a copy of the sysout containing the error documentation is sent to the requestor to help him determining the cause of the problem. In that case, BCV5 tasks are kept, to allow manual restart the morning after, once the problem cause has been fixed.
- Tasks are deleted from the BCV5 repository and from TWS at the end of the process, only in case of successful completion of the task.



6.

**New functions -
on going**



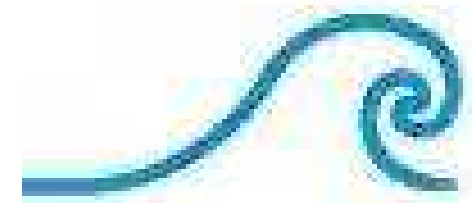
New functionalities, already active

Due to the success of the new copy application, we decided to extend it for the people of the QA area needs.

Multiple copy

They need to run tests over several DBs whose data are related each other

- Need to copy all DBs in the same night to guarantee data congruency
- We introduced the concept of 'Application'. Each application refers to a list of databases, kept in a DB2 table. Requestors don't need to know the names of the DBs in the list.

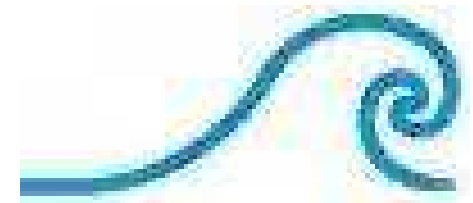
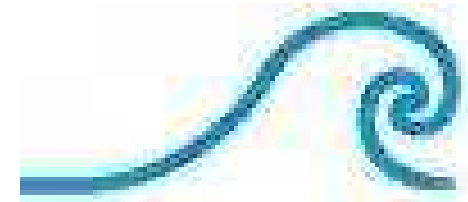


New functionalities, already active

- Copy request by Application (several databases)
- Different subject in the email:

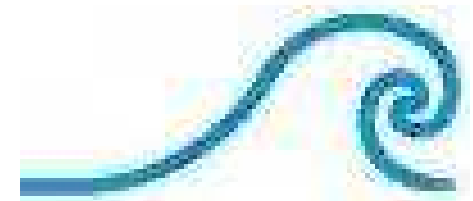
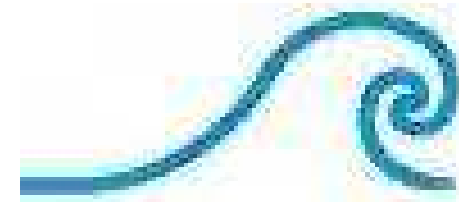
COPIAPLUS for a multiple database request

- Body must contain the following parameters:
 - The name of the application: in this case, a DB2 table is queried to get the names of the databases belonging to that application
 - The subsystem name of the source environment
 - The subsystem name of the target environment



New functionalities, already active

- If the request concerns an application, a different REXX program is started
 - Reads a DB2 table keeping track of Requestors authorized to issue those requests (typically, to copy application data spreading across multiple databases)
 - Builds the list of all DBs to be copied
 - Each database of the list is then treated as a single database request, therefore using the same flow of the single database request
- If the request is accepted, a several BCV5 tasks are using the BCV5 batch interface
- Email notification is sent to both requestor and database owner



New functions on going

New email subjects will be recognized by Mail2ZOS to provide:

- COMPARE between source and target database to discover the differences in the DDLs, running BCV5 Stage1 and Stage2 immediately, without scheduling anything overnight
- DROP target database (only in the test environment), to align DDLs to those of the source, with options to keep existing target GRANTS, SYNONYMS and BIND of the target environment
- KILL the copy request both at database and application level.



