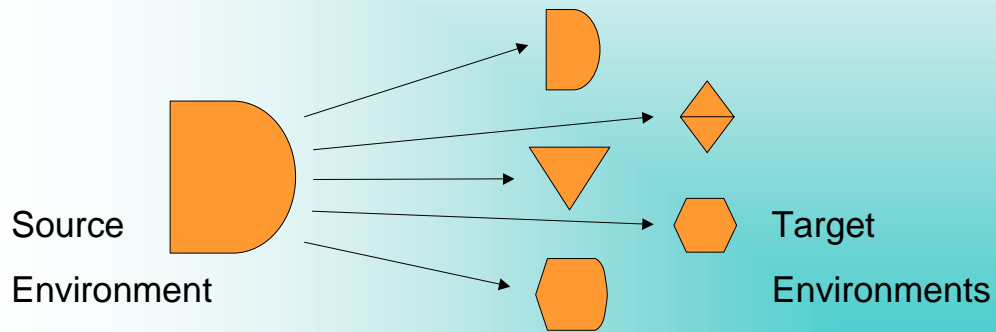


HotCopy

Data Propagation

Propagation = automatic and steady supply of heterogeneous target environments with relevant data from source environments



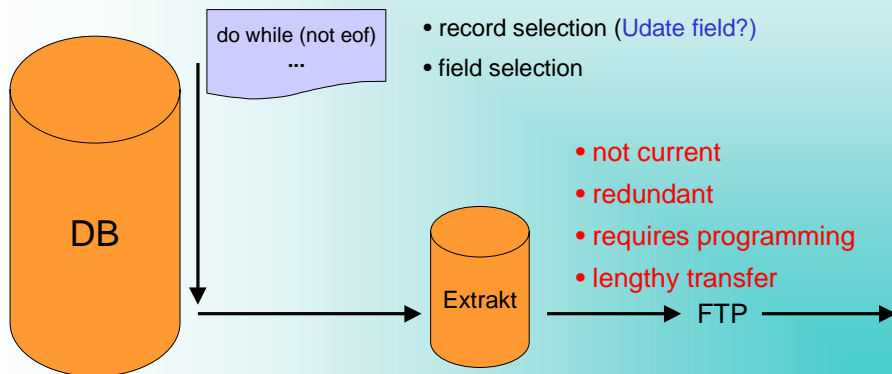
(c) UBS Hainer GmbH 2001

HotCopy is a system designed to automatically supply heterogeneous target environments with data from central sources. This way of supplying data we call *propagation*.

HotCopy

The way it was ...

Batch JOB for extraction ... & FTP..



(c) UBS Hainer GmbH 2001

There are currently various methods to transfer data from one application to another and each has its advantages and disadvantages, take a closer look at some:

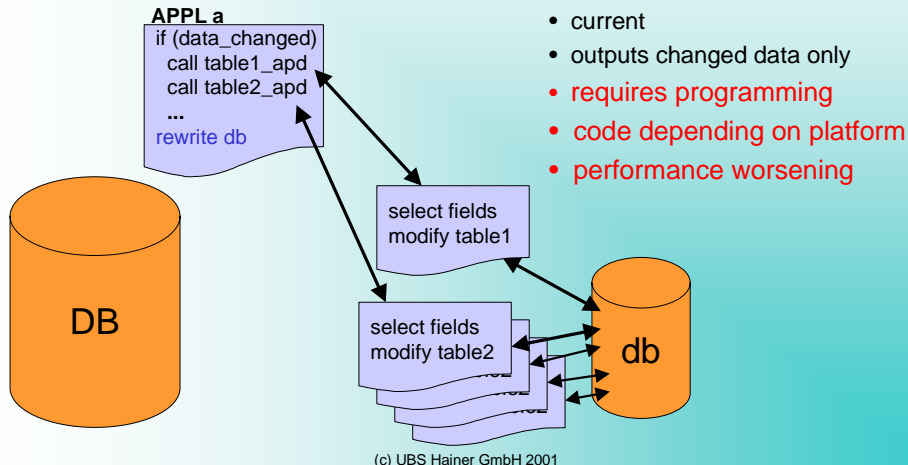
1. A regular **copy of the complete source data** and recreation of the target data, for instance, every night everything is copied.
 - Depending on the size of the source database a considerable amount of data will have to be moved. This disadvantage increases proportionately to the distance of the target application.
 - The target data will not be right up-to-date since the amount will restrict frequent transfers.
 - If the transfer is interrupted the complete process must be repeated (lack of recovery).
 - Without a suitable extraction tool the complete database will have to be transferred, even if only certain fields are necessary.

2. Regular **transfer of the changed data** and update of the target data after an initial load.
 - The amount of data to be transferred will be minimized.
 - How up-to-date the data in the target environment are, depends on the frequency of transfers.
 - One problem is, how to recognize changes. Usually this will mean the introduction of update flags into the source data.
 - Without a suitable extraction tool reprogramming will be required to support the update selection logic.

HotCopy

The way it was ...

Change the application



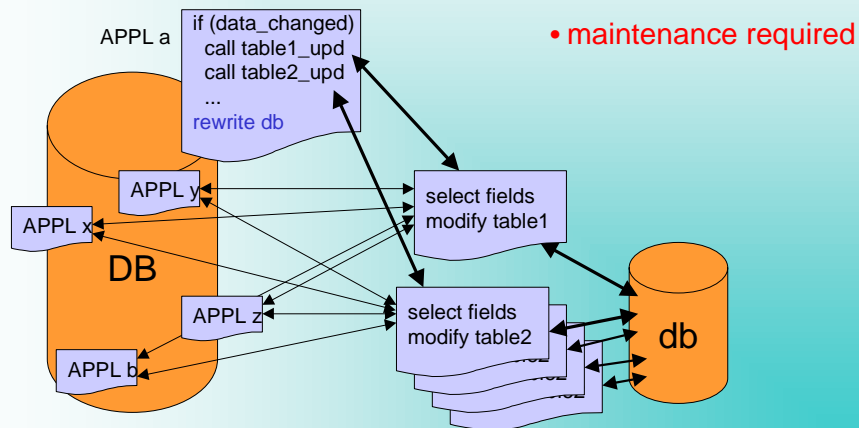
Reprogramming the source application to transfer the updates directly to the target.

- Data selection can be fitted match requirements exactly.
- The transfer of data to the target is almost real-time.
- Demands significant programming effort: risks when changing an existing application and, of course, constant maintenance.
- Performance penalties in the source application can hardly be avoided since during a synchronous transfer the source application has to wait on confirmation from the target. (Commit may be made only when the transaction is complete.)

HotCopy

The way it was ...

Change the application



(c) UBS Hainer GmbH 2001

Reprogramming the source application to transfer the updates directly to the target.

- Maintenance is an issue
- particularly, when there are after time many data objects to be transferred.

HotCopy

The way it was ...

Trigger

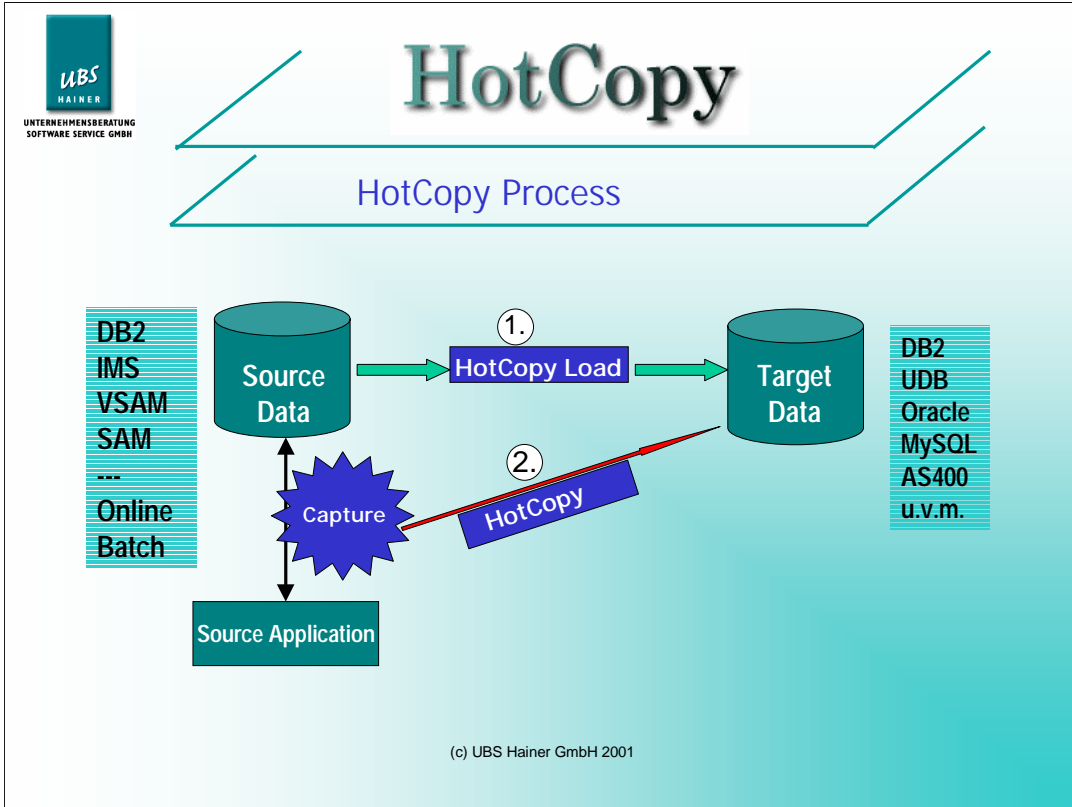
„Trigger-based techniques typically insert incremental updates to a staging table residing in the same DBMS as the primary database. In many cases, an update is written for each changed field within a record.“

Trigger are not available for IMS, DL/1, VSAM

(c) UBS Hainer GmbH 2001

Using so called triggers

- Triggers are available only in certain environments (DB2, Oracle)
- A trigger is a form of synchronous transfer and therefore slows down the source application. Triggers between two DB2 systems, for instance, will have a significant negative impact on the performance of both systems.



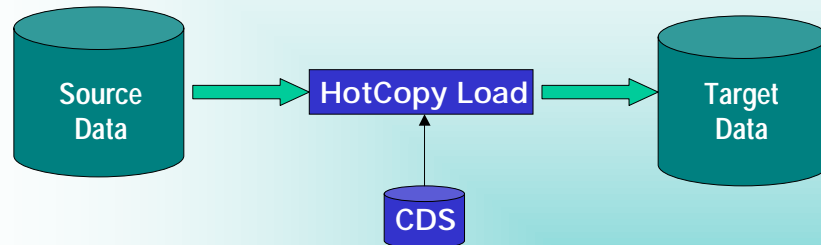
The basic concept of HotCopy is that data only need to be loaded once into a target database (HotCopy Load) and thereafter only the incremental changes (*deltas*) need be added in chosen intervals (HotCopy).

Unlike most ETL (Extract Transform Load) tools HotCopy does not extract source data after processing but instead observes the change process and notifies the change to the target database.

In the following we refer to the application where the data are created as the *source application* and the data it creates as *source data*. The receiving application is referred to as the *target application* and its data are the *target data*. The process of propagation means diverting a copy of selected data from the source application and directing it to the target application for processing.

HotCopy

HotCopy Initial Load

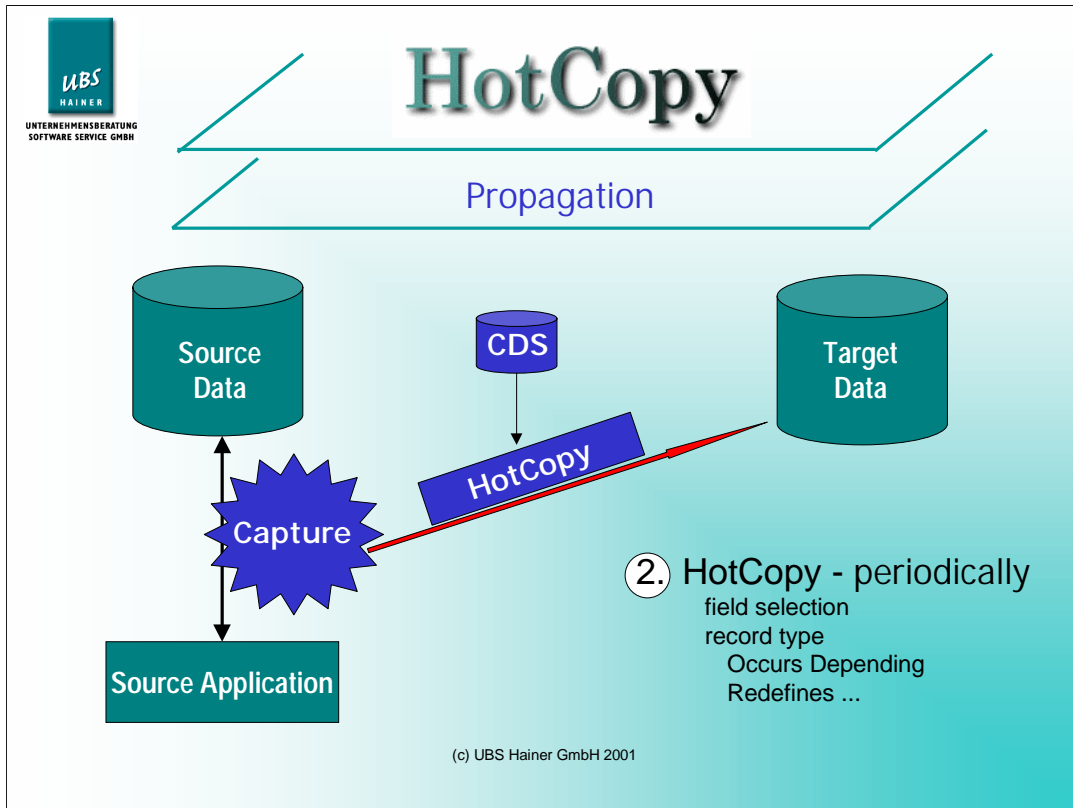


1. Initial Load - onetime process

field selection
 record type
 Occurs Depending
 Redefines ...

Key

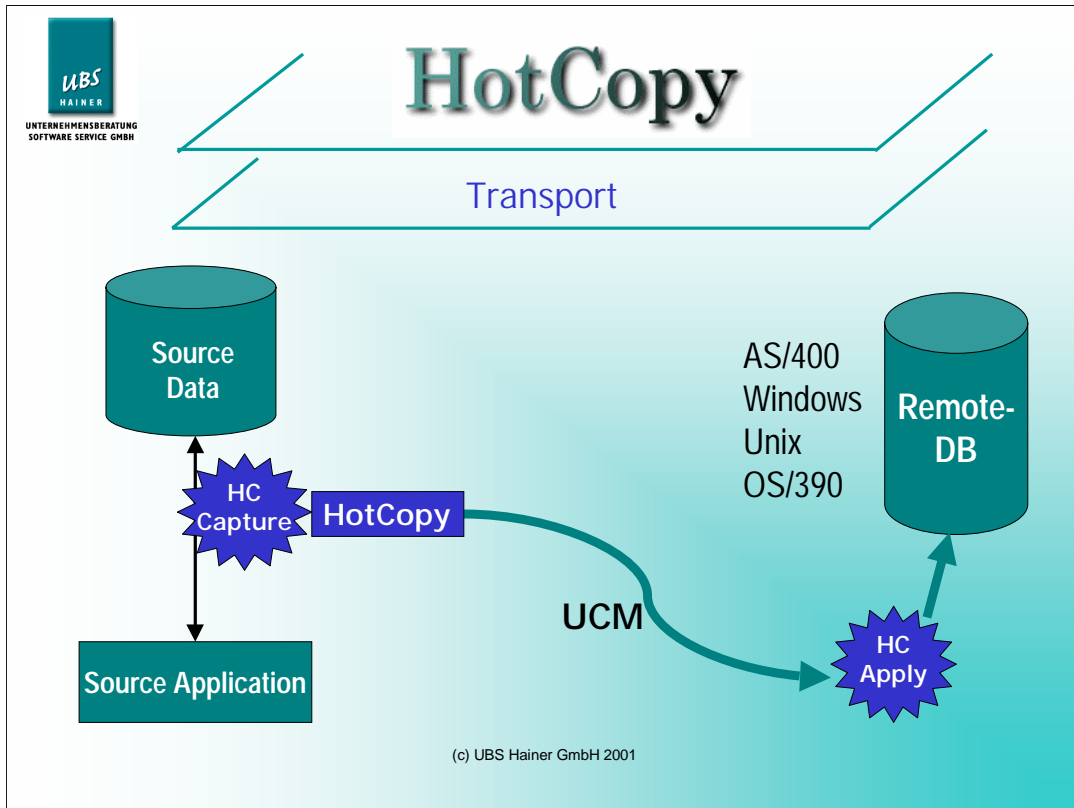
The first step is a **HotCopy initial load**. Here the desired attributes/fields of the source data are carried over to the target. After this one-time initial load the permanently active HotCopy propagation process, the **HotCopy** task, is started. HotCopy now ensures that relevant changes in the source data which occur after the initial load will be passed on to the target database. The user specifies the relevant fields of a record (VSAM) or of a segment (IMS) or columns/attributes of a table (DB2) for the initial load and for HotCopy as so called **HotCopy objects**.



After creation of the target database HotCopy will ensure that all relevant changes which are made in the source data are also mirrored in the target.

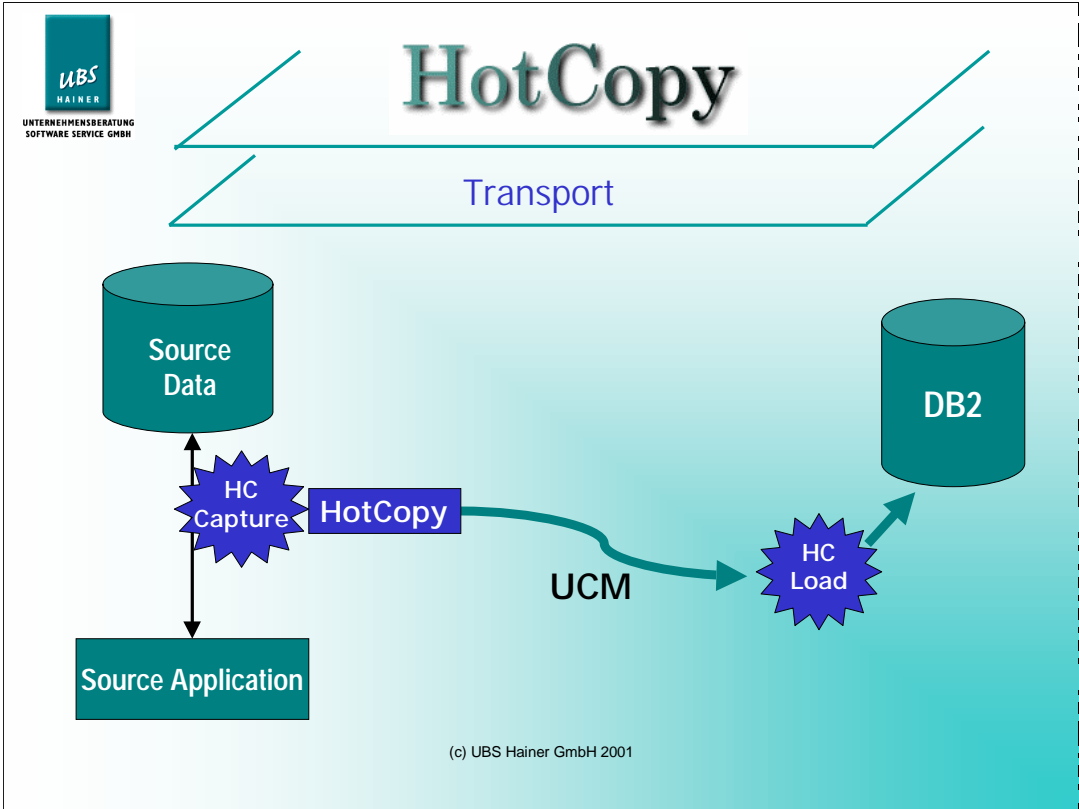
For performance HotCopy works asynchronously but ensures that only consistent (committed) logical data are propagated.

The HotCopy process guarantees current status of target data and avoids the above mentioned disadvantages of other methods: poor performance, extensive reprogramming, constant maintenance and repeated, redundant transfers of mass-data.

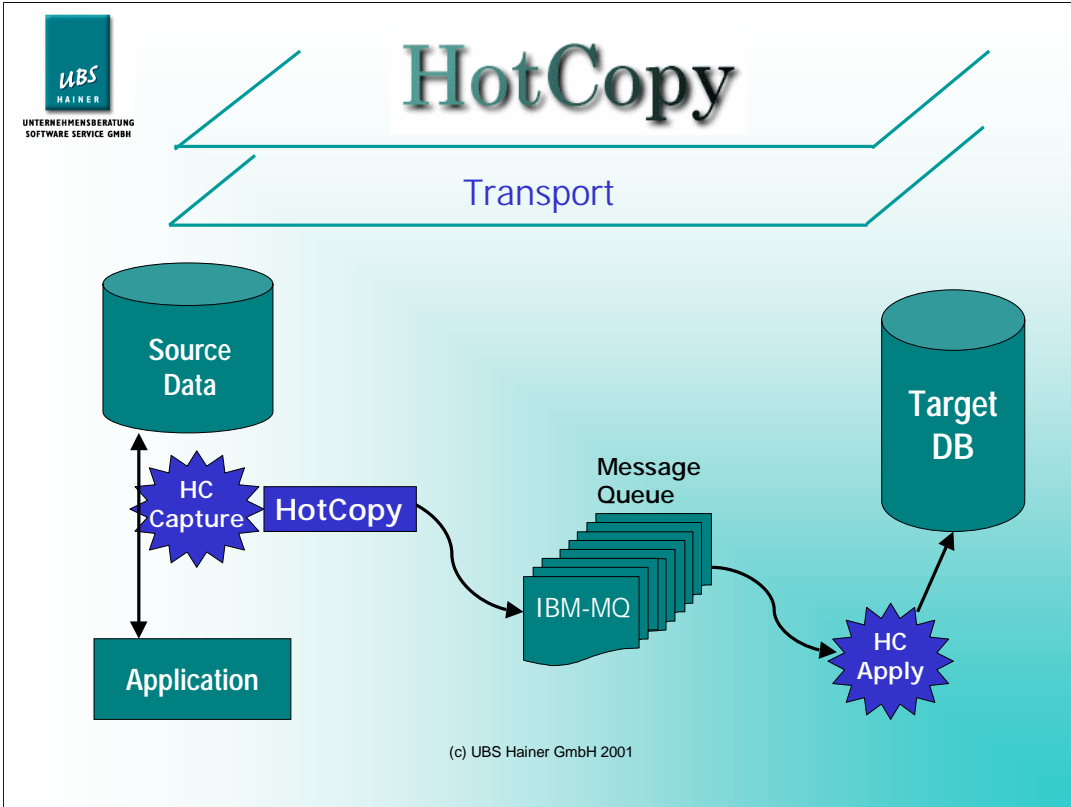


HotCopy supports various environments on source and target side. For data transmission from one system to another HotCopy uses UCM (UBS Command Manager), which is totally integrated into HotCopy.

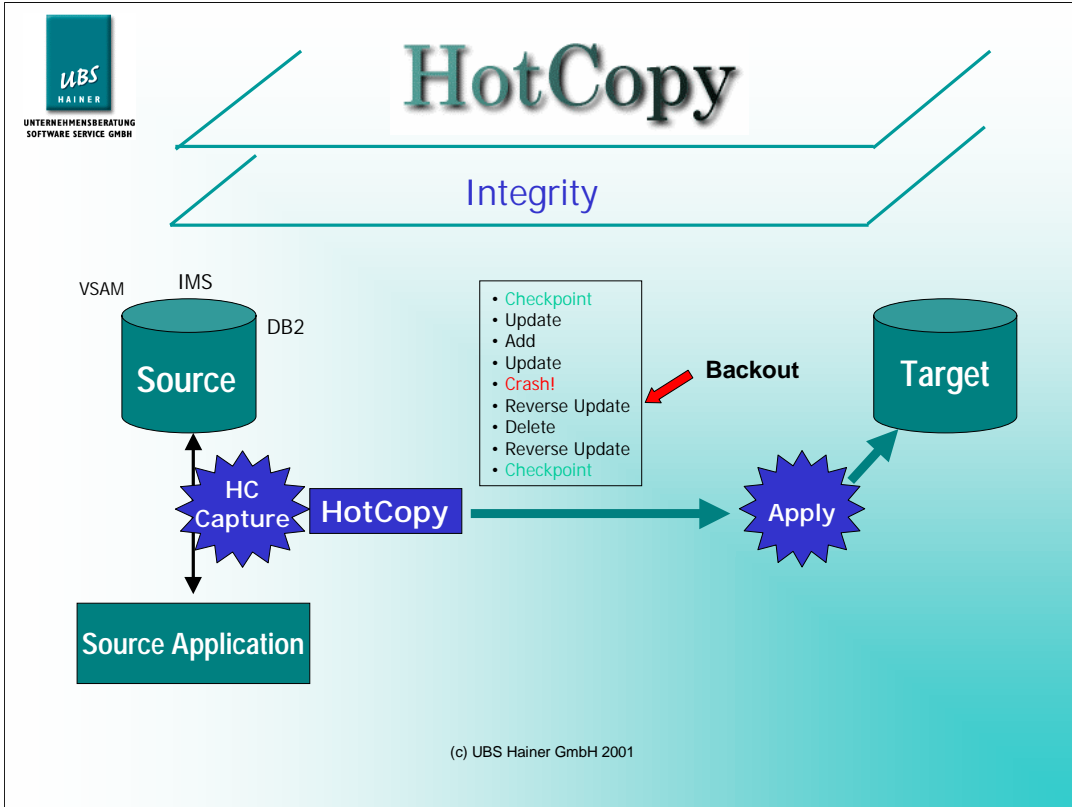
UCM initiates and controls HC Apply. There are two different apply programs, a sql-based program and a DB2 loader program.



HC Load for DB2



HotCopy can use IBM's MQ-Series.



HotCopy's Recording facility (HC Capture) sends messages of changes (INSERT, DELETE, UPDATE, CHECKPOINT, COMMIT) by way of Cross Memory Call (XMS) to the address space of the HotCopy Started Task. The same way HotCopy receives messages about failures i.e. Reverse Update, Backout.

- HotCopy ensures that only consistent (committed) logical data are propagated.

HotCopy

Hotcopy, how it works

Steps of Definition

1. Name the HotCopy Object and choose a Layout you have

```
Object Name: KDEXP1  
  
Input Object      ==> DSN1.ROLAND.KDSTAMM  
Input Object Type ==> 3  
1 - PLI Include  
2 - Cobol Copy  
3 - DB2 Table  
4 - IMS DBD  
5 - IMS PSB  
6 - Assembler Include
```

(c) UBS Hainer GmbH 2001

HotCopy Objects

With the help of an ISPF menu-controlled the user defines the tables, database segments and files to be controlled and the data to be propagated. These definitions are known as **HotCopy objects**.

HotCopy objects are maintained in the **HotCopy Database (CDB)**. A HotCopy object consists of:

- A unique name,
- A description of the source data, in this case the complete record or segment layout of the database concerned. This description can have a different form, dependent on the original database. Thus in the case of DB2 it will be taken from SYSTABLES. In the case of IMS we would use the IMS internal DBDs and PSBs or alternatively the COBOL/Assembler copy books or PL/1 includes.

HotCopy

Hotcopy, how it works

Steps of Definition

1. Name of HotCopy Object

2. Select attributes/fields to be propagated/loaded

S	LV	Field Name	Typ	Length	
	1	KDSTAMM	S	0	
	2	DEC010	D	10	
	2	FLDABC	I	2	
	2	FLDDEF	I	4	
K	2	Q4711	I	4	key
S	2	NAME1	C	8	selected
S	2	VNAME1	V	10	selected
S	2	NAME2	C	8	selected
	2	INDTE	A	4	
	2	INTME	T	3	
	2	TMEST	M	10	

(c) UBS Hainer GmbH 2001

- A definition of the fields/attributes to be selected. The user selects these from a panel which shows all fields of the database.
- A specific key for the target table in which the user marks the corresponding data fields as key fields.

HotCopy

Hotcopy, how it works

Steps of Definition

1. Name of HotCopy Object

2. Select Fields/Attributes

3a. Specify conditions, if necessary

```
HCP-Object . . : KDEXP1  
  
Structure . . . . . : KDSTAMM  
  
Selected condition field : FLDABC  
Condition field length . : 2  
Condition field type . . : I  
  
Compare string ==> 22
```

(c) UBS Hainer GmbH 2001

- Clauses to control handling of non-relational originating elements – in COBOL or PL/1 there are often data structures which are simply not relational: REDEFINE arrays, OCCURS DEPENDING and the like. HotCopy is capable of propagating such structures into chained relational tables.

HotCopy

Hotcopy, how it works

Steps of Definition

1. Name of HotCopy Object

2. Select Fields/Attributes

3b. Specify extended rules via REXX, if necessary

```
EDIT HCP Extended Logic  
***** Top of Data *****  
  
000001 arg data_1  
000002 if ( rectxp = „1“ ) then return 1  
000003 return 0
```

(c) UBS Hainer GmbH 2001

- Definition of the conditions which trigger propagation – it is possible make the propagation dependent on complex conditions based on the contents of multiple fields.

HotCopy

Hotcopy, how it works

Steps of Definition

1. Name of HotCopy Object

2. Select Fields/Attributes

3. Define Conditions, if necessary

4. Specify target

```
Object Name:      KDEXP1
Object Dataset Name    ==> DSN1.ROLAND.KDSTAMM

Local Host Collection Datasets :
Dataset Output        ==> GERD.TESTOUT4

Remote Host Collection:
Remote Host IP Address ==>

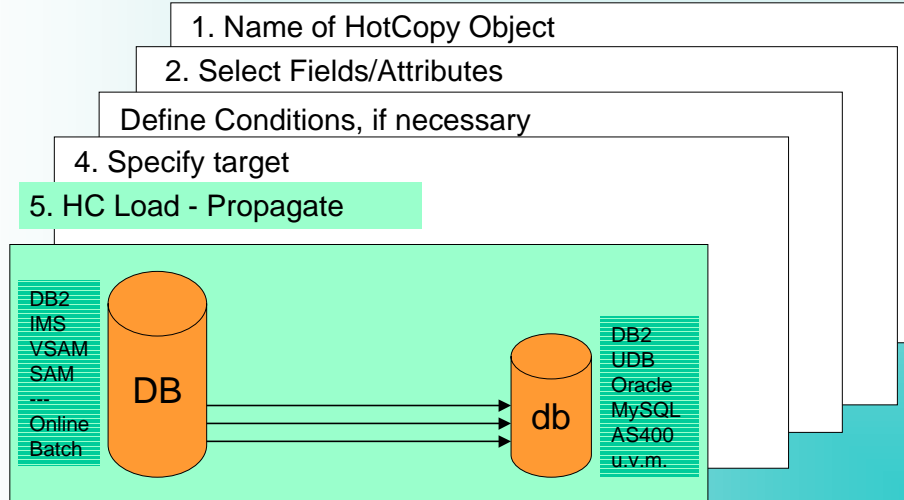
MQ Series interface :
Queue Manager         ==>
Queue Name            ==>
```

- Definition of the target system for propagation: TCP/IP address or name, port, database, table name, etc.

HotCopy

Hotcopy, how it works

Steps of Definition



Before HotCopy can start propagation, a HotCopy initial load (HC LOAD) must be run. HC LOAD uses the information stored in the affected object to determine source, target and transformation required to create one-time a target. HC LOAD ends with a “loaded flag” for the affected object and HotCopy starts with propagation.

Finally HC LOAD sets a LOADED flag in the relevant HotCopy object. This signals the start of the propagation process to HotCopy. When started, the HotCopy task (HCP) will commence propagation for all objects in the CDB which have the LOADED flag set.

HotCopy

Hotcopy, how it works

```
01 REC.  
03 REC-KNZ PIC X. ← Record type: REC-KNZ = „A“  
03 REC-A.  
05 DFLD0001 PIC X(30).  
05 DFLD0002 PIC X(50).  
05 FILLER PIC X(2).  
03 REC-B REDEFINES REC-A.  
05 DFLD0005 PIC X(30).  
05 TAB-1 OCCURS 1 TO 10 DEPENDING TAB-CNT.  
10 TAB-ELEM1 PIC X(2).  
10 TAB-ELEM2 PIC 9(3).  
05 TAB-CNT PIC 9(2).
```

only
DFLD0001,
DFLD0002

1 output record per table element
(make relational)

HotCopy

Occurs --> relational structure

```

COBOL-COPY.
01 KD.
  03 KDNR ...
  03 KDORT ...
  03 KD-REC.
    05 TAB-CNT PIC 9(2).
    05 TAB-1 OCCURS 1 TO 10 DEPENDING TAB-CNT.
      10 ELEM1 PIC X(2).
      10 ELEM2 PIC 9(2).
  
```

KDNR	KDORT	KDNR	TAB-CNT	ELEM1	ELEM2
4711	Fulda	4711	01	BC	53
4712	Bielefeld	4711	02	CC	22
4713	Saarlouis	4711	03	SF	71
4714	Hopfmansfeld	4712	01	AA	17

(c) UBS Hainer GmbH 2001

HotCopy

Redefine --> relational structure

```

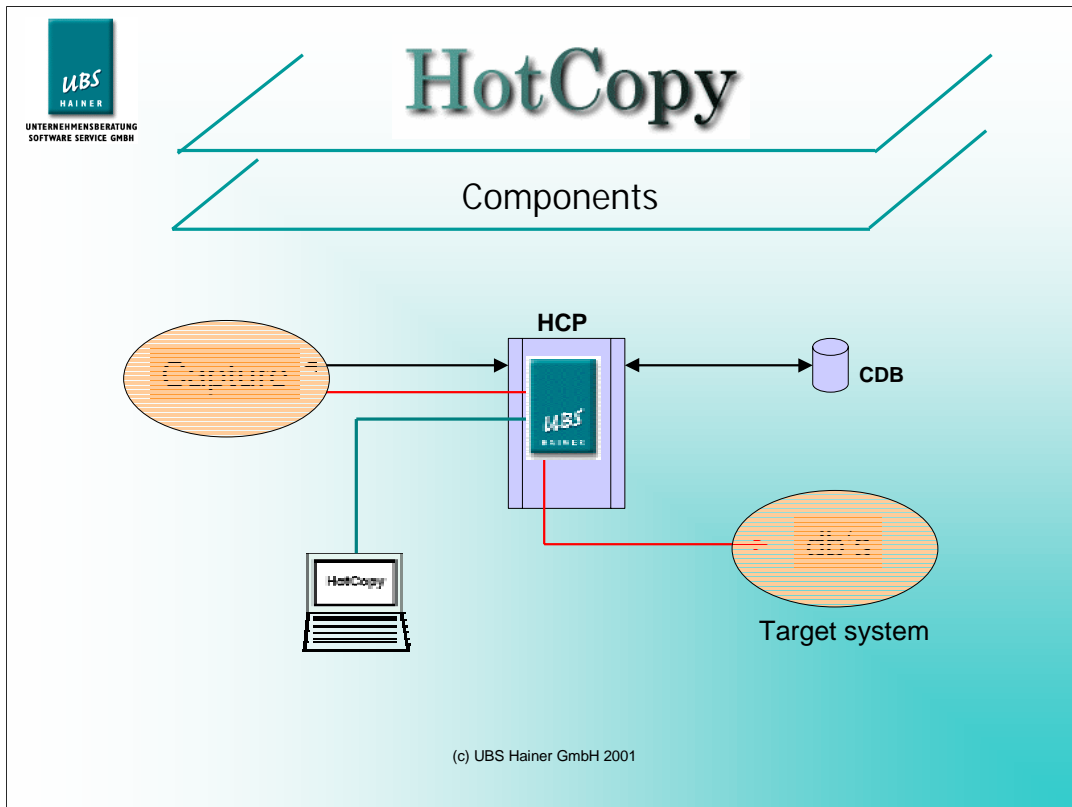
COBOL-COPY.
01 KD.
03 KDNR ...
03 KDKZ ...
03 KD-REC.
05 KDDATE ...
05 KDDORT ...
03 KD-REC-UMS REDEFINES KD-REC.
05 UMSATZ-LFD ...
05 UMSATZ-VJ ...
05 UMSATZ-VVJ ...
  
```

HotCopy-Objekt 1

KDNR	KDNAME	KDORT
4711	Meier	Düsseldorf
4712	Müller	Frankfurt

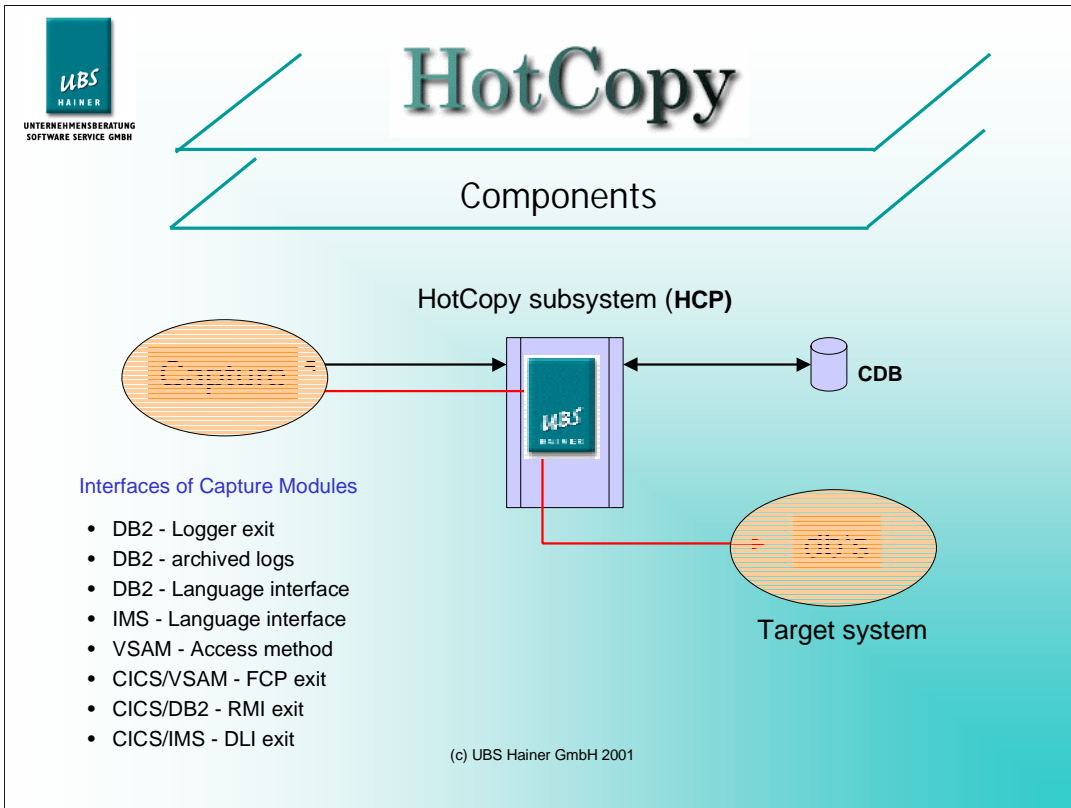
HotCopy-Objekt 2

KDNR	UMS-LF	UMS-VJ	UMS-VV
4711	100.000,-	98.000,-	77.000,-
4712	210.000,-	190.000,-	50.000,-



Technically HotCopy consists of:

- Various data Recording facilities (capture modules) to recognize a change of data in its respective environment, IMS, CICS or Batch, and to report it to the HotCopy started task (HCP).
- The HotCopy Started Task (HCP) which collects the data from the data Recording facilities in Dataspaces, selects where appropriate and transfers them to the targets.
- A data transfer facility which transfers consolidated change data to the target application and updates it.
- An ISPF user interface to control HotCopy and maintain the HotCopy database.
- The HotCopy database (CDB) which contains all data structures (HotCopy objects) to be propagated by HotCopy.



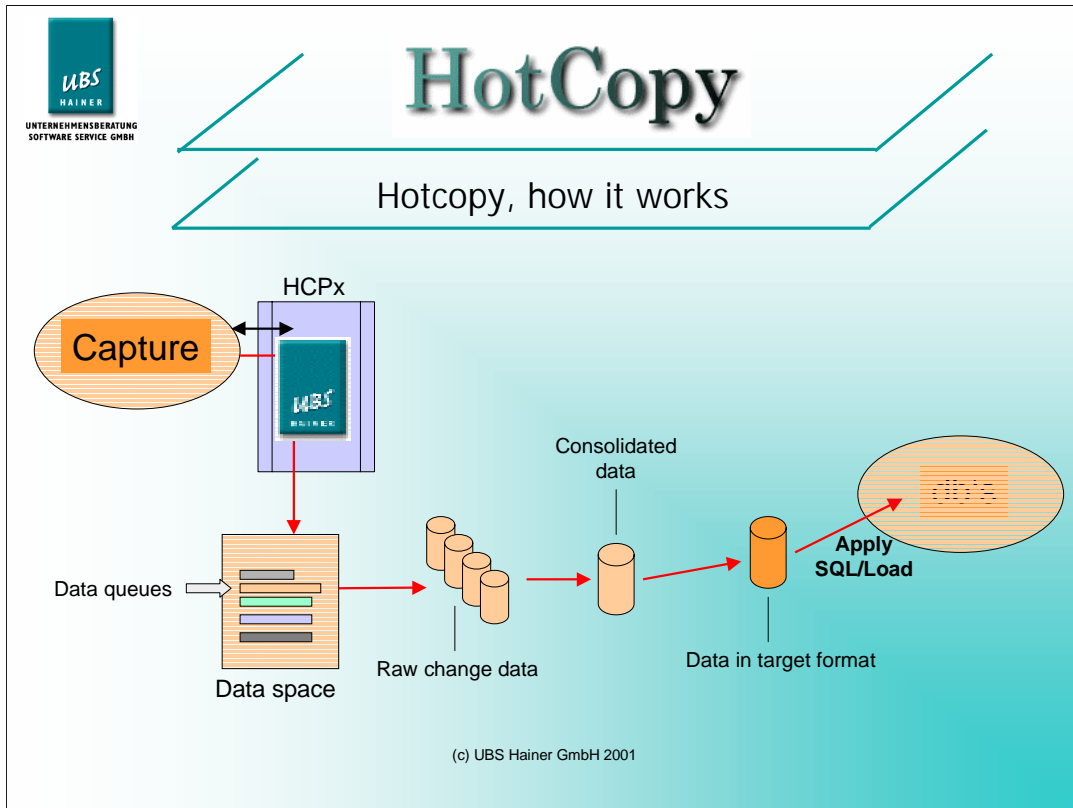
The following source and target database formats are supported:

Source formats:

- DB2 databases on OS/390.
- IMS databases
- VSAM datasets.
- Sequential files.
- The above under IMS, CICS and in batch.

Targets:

- DB2 databases for OS/390.
- Any SQL database system, e.g. Oracle, UDB, MySQLm DB2/400.



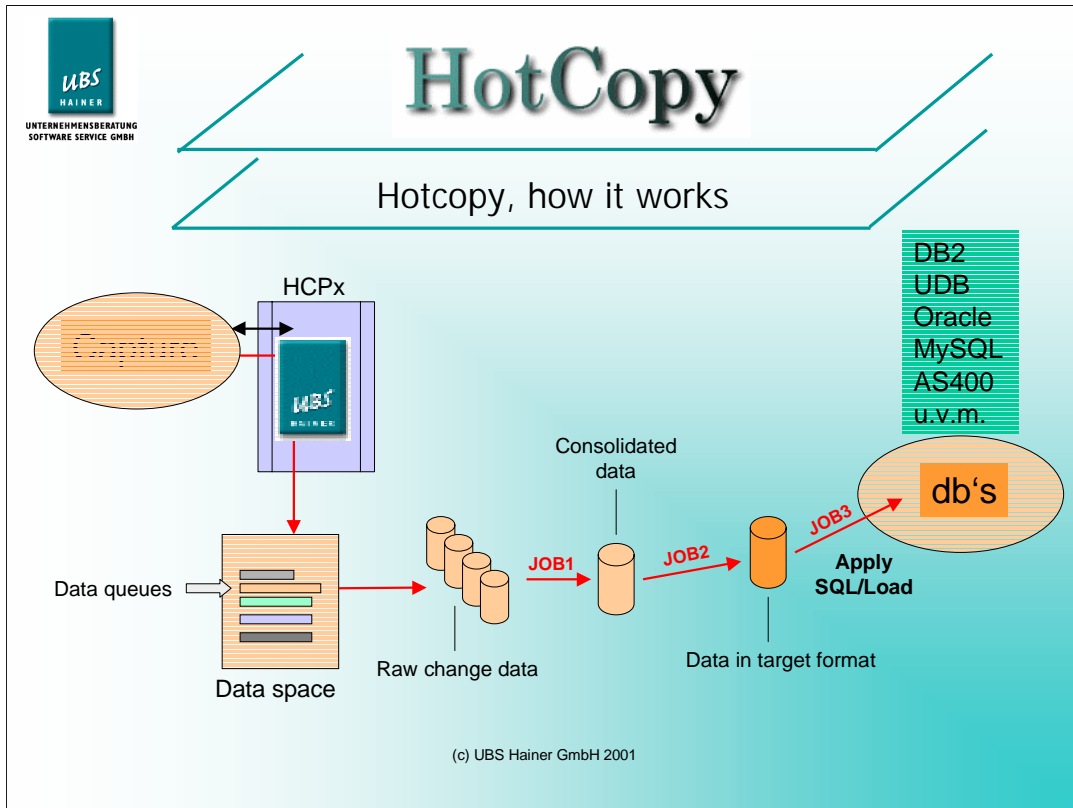
After creation of the target database HotCopy will ensure that all relevant changes which are made in the source data are also mirrored in the target. The HotCopy components work together as follows:

- HotCopy's Capture facility sends messages of changes (INSERT, DELETE, UPDATE, CHECKPOINT, COMMIT) by way of Cross Memory Call (XMS) to the address space of the HotCopy Started Task (HCP). Depending on the environment the Recording facility agents are placed in various interfaces:
 - ✓ DB2 Logger Exit or Archived Log
 - ✓ IMS Language Interface
 - ✓ VSAM/SAM Subsystem Interface
 - ✓ CICS File Control Interface (VSAM), DL/I Interface

The HotCopy Started Task (HCP) manages the delta changes reported by the Recording agents in a data space. For each active HotCopy object there is a data queue reserved therein. The data space is persistent, that is, it will survive a restart of the HotCopy address space or a restart of the operating system.

After a predefined interval or on an operator/administrator command or when a data queue reaches its threshold at the latest, a CLOSE event may be triggered for any one of the data queues. This causes the system to seek a logical termination point, that is, only data from currently running transactions will be taken into the Level-1 repository, data from newly started transactions will remain in the data space for the time being.

As soon as this activity has completed, a job will be started which turns the Level-1 repository into a level-2 repository, by sorting out the irrelevant transactions. The



data are still raw changes. In Level-1 there may be more than one file since a new one is set up every time a data queue reaches a defined threshold. In this case the job would process several, concatenated input files.

After successful completion of this process the data will be handled by a second batch job. "job 2", which is triggered automatically on completion of the predecessor and converts the data to the corresponding format required by the target system. This job creates a Level-3 repository and triggers a third batch job.

The third batch job handles the transfer to the target system and the update of the target data. For transfers to workstations and other non-MVS systems the UCM-UBS Command Manager is used. For mainframe DB2 a modified load utility is utilised.

Each of these jobs is monitored during its run by the HCP and its history recorded. Abended jobs are flagged and can be restarted individually (from the HCP panels). Besides that, no new Level-3 job will be submitted for the same repository thread when a predecessor has failed.

HotCopy

Major Benefits

- Only "deltas" are transferred - even so update fields are not required in source data - changes are captured
- Up-to-dateness of target data is adjustable: daily, hourly, ..
- "Make relational": Record types (VSAM), segment chains (IMS) will be spread over consistent tables
- Long running FTPs avoided
- Data Warehouse, WEB-Applications, etc. have access to up-to-date host data without disturbing host applications
- Programming and maintenance avoided, no risk to damage or degrade source application
- Any number of targets, items of source and target may be named different