

FACILITATING DATA MIGRATION TO MAINTAIN CONTINUOUS INFORMATION AVAILABILITY IN THE MEDIUM SCALE ORGANIZATION

Prof. Sachin Patil

Asst. Professor MCA Dept. KES's PIBM, Chinchwad, Pune

sachin.pibm@pratibhagroup.org.in

Abstract:

The rapidly increasing volumes of data generated by businesses can create significant data management challenges in Medium scale industries. When responding to business demands or compliance, companies must be able to organize and access volumes of data stored in a variety formats. Maintaining legacy devices- and the equipment necessary to read them - can be expensive and onerous. In addition, because device storage has a high malfunction rate, data recovery is often required in order to refurbish the data prior to migration. Data migration is necessary when a company upgrades its database or software, from one version to another or from one application to an entirely different application. Software or applications can be particularly written, using entire programs, to ease data migration. Such program switches or upgrades can take place as a result of normal company practices or as a result of directives mandated in the wake of a company takeover. Data Migration is the process of transferring data between storage formats, types, devices or computer systems. It is often required when organization change computer systems or upgrade to new systems. This solution is usually performed programmatically to achieve an automated migration. As a result, legacy data stored on out of date or obsolete formats is assessed, indexed, re-duplicated and then migrated to newer more cost effective and consistent storage media. So enabling data migration will support continuous data availability for decision making in medium scale industries

which are likely to change.

Keywords: Data Migration, Storage device, Media, Data Cleansing, Data availability.

Introduction:

Data migration means what it sounds like it means -- sort of. It's not data that moves one from place to another, unless you think of places as being virtual. Data migration is actually the translation of data from one format to another format or from one storage device to another storage device. This also necessarily requires someone or something to do the translating. Data doesn't just get up and walk to another format all by itself.

Data migration is necessary when a company upgrades its database or system software, either from one version to another or from one program to an entirely different program. Software can be specifically written, using entire programs or just scripts, to facilitate data migration. Such upgrades or program switches can take place as a result of regular company practices or as a result of directives mandated in the wake of a company takeover. Another use of data migration is to store little-used data on magnetic tape or other backup storage methods. This data may need to be stored for historical purposes or for periodic access. Individual computer users do this all the time when they back up their data to CDs, DVDs, or external hard drives. Companies large and small do this, of course, to protect and archive their data. Migrated data typically is moved offline but remains available via network access, leaving the online environment free to conduct current business.

Need:

The rapidly increasing volumes of data generated by businesses can create significant data management challenges. When responding to business demands or compliance and litigation requirements, companies must be able to access and organize volumes of data stored in a variety of formats. Maintaining legacy devices - and the equipment necessary to read them - can be costly and burdensome. In addition, because device storage has a high malfunction rate, data recuperation is often required in order to refurbish the data prior to migration.

Tape recovery and data migration are one of the most complex activities to manage, predict and perform. Tape media formats need to be read in their native states and imaged to a device which permits low level edits in order to gain access to the information.

Media with physical damage and misaligned write errors all have significant complexity associated with gaining access to the information. The particular agent used and the type of data stored further adds complexity to the scope of the work. A well designed migration methodology ensures that your data is properly evaluated, reviewed and restored before it is migrated to new, more accessible and cost-effective media. And, the process makes certain that your valuable data is safeguarded from beginning to end.

Achieve Data migration:

To achieve an effective data migration procedure, data on the old system is mapped to the new system providing a design for data extraction and data loading. The design relates old data formats to the new system's formats and requirements.

Programmatic data migration may involve many phases but it minimally includes data extraction where data is read from the old system and data loading where data is written to the new system.

If a decision has been made to provide a set input file specification for loading data onto the target system, this allows a pre-load 'data validation' step to be put in place, interrupting the standard E(T)L process. Such a data validation process can be designed to interrogate the data to be transferred, to ensure that it meets the predefined criteria of the target environment, and the input file specification. An alternative strategy is to have on-the-fly data validation occurring at the point of loading, which can be designed to report on load rejection errors as the load progresses. However, in the event that the extracted and transformed data elements are highly 'integrated' with one another, and the presence of all extracted data in the target system is essential to system functionality, this strategy can have detrimental, and not easily quantifiable effects.

After loading into the new system, results are subjected to data verification to determine whether data was accurately translated, is complete, and supports processes in the new system. During verification, there may be a need for a parallel run of both systems to identify areas of disparity and forestall erroneous data loss. Automated and manual data cleaning is commonly performed in migration to improve data quality, eliminate redundant or obsolete information, and match the requirements of the new system.

Data migration phases (design, extraction, cleansing, load, verification) for applications of moderate to high complexity are commonly repeated several times before the new system is deployed.

Some key terms in understanding data migration are:

Legacy data is the recorded information that exists in your current storage system, and can include database records, spreadsheets, text files, scanned images and paper documents. All these data formats can be migrated to a new system.

Data migration is the process of importing legacy data to a new system. This can involve entering the data manually, moving disk files from one folder (or computer) to another, database insert queries, developing custom software, or other methods. The

specific method used for any particular system depends entirely on the systems involved and the nature and state of the data being migrated.

Data cleansing is the process of preparing legacy data for migration to a new system. Because the architecture and storage method of new or updated systems are usually quite different, legacy data often does not meet the criteria set by the new system, and must be modified prior to migration. For example, the legacy system may have allowed data to be entered in a way that is incompatible with the new system. Architecture differences, design flaws in the legacy system, or other factors can also render the data unfit for migration in its present state. The data cleansing process manipulates, or cleans, the legacy data so it conforms to the new system's requirements.

Best practices for data migration:

Data migration is the process of making an exact copy of an organization's current data from one device to another device—preferably without disrupting or disabling active applications—and then redirecting all input/output (I/O) activity to the new device. There are a variety of circumstances that might cause an organization to undertake a data migration, including:

- Server or storage technology replacement or upgrade
- Server or storage consolidation
- Relocation of the data center
- Server or storage equipment maintenance, including workload balancing or other performance-related maintenance.

The above scenarios are fairly routine parts of IT operations

in organizations of virtually any size. They are so routine, in fact, that more than 60 percent of respondents to a recent survey¹ indicated that they migrate data quarterly or more often—with 19 percent migrating weekly.

However, even routine processes can cause problems for IT administrators and managers. More than 75 percent of respondents to the same survey said they had experienced problems during data migration. These problems included, but were not limited to:

- Extended or unexpected downtime
- Data corruption, missing data or data loss
- Application performance issues
- Technical compatibility issues.

How can organizations minimize the business impacts of data migration—downtime, data loss and increased cost? The best way is to employ a consistent, reliable and repeatable methodology for migrations that incorporates planning, technology implementation and validation. Following image shows data migration methodology for best practices.



Data migration methodology

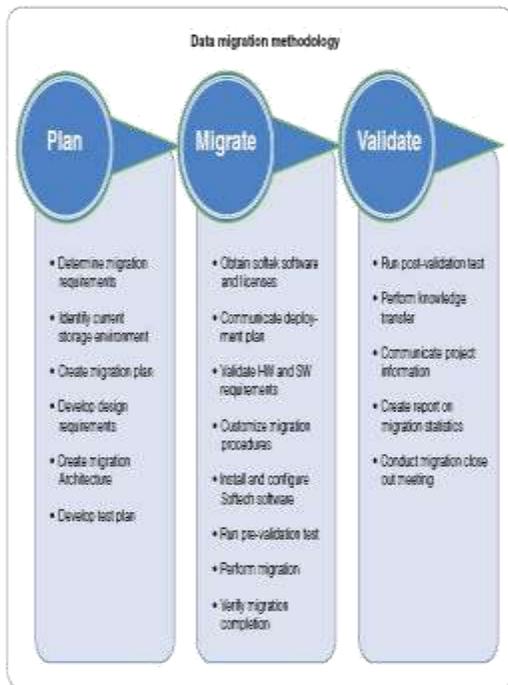


Figure 1: Migration methodology

Setting-up the Data Migration Process:

1. Choose a Data Modeling Tool with Reverse Engineering Capability,(such as ERWin).
2. Define and create the Data Dictionary.
3. Identify all the required Data Sources, and an 'owner' for each Source.
 - Data Feeds
 - Legacy Systems
 - Operational Data Stores
4. If your Sources include Access Databases, then check out MDB Tools, from Brian Burns.
5. Define the Data Items required, in consultation with the

Users.

6. Create the Data Models for the Source Data.
7. Define Data Validation Checks (bottom-up) and Clean-Up Business Rules for Source Data.
8. Carry out an Audit of the Data Quality in major Databases, (bottom-up and top-down). Evaluate the benefits of a Data Cleansing Product, such as Seamless from the C and C Group.
9. Define the Staging Area, with MIRror Tables to store Extract Files.
10. Create the Business Data Model for the Consolidated Database
11. If the final target is an ERP, such as SAP, then create the Data Model for the Target ERP Database.

For SAP, J.D.Edwards, Peoplesoft and Siebel, a Modelling Tool called Saphir, from Silwood Technology can be very useful here.

12. Define the Data Mapping between Source and Target Data Items.
13. Optionally, create a CRUD Matrix to identify the interactions between Data and Functions.
14. Define Acceptance Tests for data in the Integrated Database.

During the migration phase, the IT organization will need to communicate its plans; obtain, install and configure any necessary software; and perform the actual data migration. A premigration data validation test is recommended, in addition to post-migration validation testing. These tests confirm that the data is in the same state after the migration as it was before. Clearly, the most important part of this stage is the migration itself. As outlined above, software technology can simplify this process by enhancing the speed of migration, by minimizing or eliminating application downtime, and/or by enabling migration during regular business hours, helping the organization to get back to business as quickly as possible.

Validate:

After the migration has been completed, the IT organization should compile migration statistics and prepare a report to highlight what worked, what didn't work and lessons learned. The report should be shared with all members of the migration team. These types of reports are critical in building a repeatable and consistent process through continuous process improvement—building on what worked and fixing or changing what didn't work. Further, documenting the migration process can help train staff, and simplify or streamline the next migration, reducing both expense and risk.

Case: Researcher has done extensive work to design an application in PHP to do migration of mysql data conversion to excel data as per requirement.

Following is a sample code used to create such application:

```
<!--?php
/*
This is where we declare all the database connection
variables and their respective values */
$db_host = "localhost";
$db_user= "root"; $db_pass =
"password";
$db_link = mysql_connect($db_host,$db
user,$db_pass); $db_name = "excel";
mysql_select_db($db_name,$db_link);
/*
This section builds the required query to fetch the
data we need. The second line is calling or executing
the query, and the third line is counting the number
of fields in the database returned by the query we
built above. */
$select = "SELECT * FROM sheet1"; $export =
mysql_query($select); $fields =
mysql_num_fields($export);
/*
Here we loop and extract all the field
names from our database. */
for ($j = 0; $j < $fields; $j++) {
    $header = mysql_field_name
($export, $j). "t";
}
}
```

```

/*
In this section we are exporting the values from
database and writing them to correct columns of our excel
spreadsheet */
while($row = mysql_fetch_row($export)) { $line = "";
  foreach($row as $value) {
    if(!isset($value) OR ($value == "")) { $value = "t";
    } else {
      $value = str_replace("'", "", $value); $value = "" .
      $value . "t";
    }
    $line .= $value;
  }
  $data = trim($line)."n";
}
$data = str_replace("r", "", $data);
if ($data == "") {
  $data = "n(0) Records Found'n";
}
}
/*
In the last segment we create the header info for the
user. It uses the header() function to tell the browser
that we have a file that needs to be downloaded.*/
header("Content-type: application
x-msdownload"); header("Content-
Disposition: attachment; filename=
extraction.xls"); header
("Pragma: no-cache");
header("Expires: 0"); print
"$headern$data";
?-->

```

Web References:

- [1] http://en.wikipedia.org/wiki/Data_migration
- [2] <http://services.seagate.com/srs1-800-475-0143>
- [3] <http://www.infotech.net.org/ntca/DataMigration.htm>
<http://www-935.ibm.com/services/us/gts/pdf/softek-best-practices-data-migration.pdf>
- [4] http://www.databaseanswers.org/data_migration/general_migration_approach.htm
- [5] <http://www-935.ibm.com/services/us/gts/pdf/softek-best-practices-data-migration.pdf>
- [6] <http://forums.mysql.com/read.php>
- [7] <http://devzone.co.in/export-mysql-data-into-excel-or-csv-format/>

Conclusion:

Data migration is a routine part of IT operations in today's business environment. Even so, it

often causes major disruptions as a result of downtime or application performance problems, and

it can severely impact budgets. To prevent these problems, organizations need a consistent and

reliable methodology that enables them to plan, design, migrate and validate the migration.

Further, they need migration software that supports their specific migration requirements,

including operating systems, storage platforms and performance. In addition, migration products

that maintain continuous data availability during the migration without affecting performance are desirable.