This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the “Documentation”) is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with “Restricted Rights.” Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2013 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.
Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

If you would like to provide feedback about CA Technologies product documentation, complete our short customer survey, which is available on the CA Support website at http://ca.com/docs.
Contents

Chapter 1: Overview  ................................................................. 7
1.1 Key Concepts ................................................................................. 10
1.2 Major Features .............................................................................. 14

Chapter 2: COST ACCOUNTING CONCEPTS FOR IS ORGANIZATIONS  25
2.1 Essential Characteristics of an IS Accounting  ................................. 28
2.2 Budgeting Concepts ....................................................................... 31
  2.2.1 What is a User Budget? ............................................................... 32
  2.2.2 Budget Detail, How Much is Enough? ........................................... 34
  2.2.3 Steps in Developing a Budget ....................................................... 35
  2.2.4 Reporting Budget Information .................................................... 38
2.3 Resource Accounting ..................................................................... 40
  2.3.1 CPU Time ............................................................................. 42
  2.3.2 CPU Time/Memory Utilization Accounting ................................. 42
  2.3.3 CPU Occupancy Time ............................................................... 43
  2.3.4 Transaction Residency Time ....................................................... 45
  2.3.5 Resource Occupancy Time ......................................................... 46
2.4 Transaction Accounting .................................................................. 47
  2.4.1 Resource Accounting/Transaction Accounting Examples .............. 48
  2.4.2 Implementing Transaction Accounting ........................................ 50
  2.4.3 Summary of Transaction Accounting ......................................... 61
2.5 Differential Charging .................................................................... 62
  2.5.1 Priority Accounting .................................................................. 63
  2.5.2 Zone Accounting ...................................................................... 64
  2.5.3 Product Surcharge Accounting .................................................. 65
2.6 Additional Accounting Options ..................................................... 66
  2.6.1 Support Services Accounting ..................................................... 67
  2.6.2 Billing Options ...................................................................... 68
  2.6.3 IS Inventory and Assets Accounting ........................................... 69
2.7 DASD/DFHSM/SMS Storage Accounting ........................................ 70
  2.7.1 Permanent DASD Space Accounting ......................................... 72
  2.7.2 Temporary DASD Space Accounting .......................................... 74
  2.7.3 DFHSM Migrated Data Sets ....................................................... 75
  2.7.4 DFHSM Backup Data Sets ......................................................... 77
  2.7.5 VSAM Accounting .................................................................. 79
  2.7.6 System Managed Storage (SMS) Accounting ............................ 81
Chapter 1: Overview

CA MICS Accounting and Chargeback Option is a revenue management and resource consumption control system designed to meet the accountability requirements of information systems (IS) organizations.

With data processing generally accepted as an indispensable part of business operations, organizations today hold IS management teams responsible for making sound business decisions in response to technical situations. This responsibility places an increasing demand on the IS team to ensure effective use of information technology investments and to characterize performance and capacity issues in financial terms. The ability to quantify, in monetary terms, the value of individual services supplied to users of various information systems and facilities is of major importance to IS management.

The standard features of CA MICS Accounting and Chargeback Option give you maximum flexibility to meet the chargeback requirements of your installation, to respond to technological change, and to establish the basis for certain aspects of enterprise financial reporting, budgeting, and forecasting. In addition, because its reports provide concise accounting information in monetary terms that everyone can understand, CA MICS Accounting and Chargeback Option promotes improved financial communication and awareness for everyone from users to top-level management.

CA MICS Accounting and Chargeback Option is part of the CA MICS IS Management Support System (see Figure 1-1). CA MICS is a comprehensive, flexible application system that applies standard management practices to the IS organization. It provides integrated applications analogous to the integrated financial applications that are now indispensable to enterprise financial management.

CA MICS is comprised of Data Integration Applications and Management Support Applications (one of which is CA MICS Accounting and Chargeback Option). Data Integration Applications validate, interpret, consolidate, and format data from diverse sources and locations and store the subsequent information in the CA MICS database. Management Support Applications process the information in the CA MICS database in support of IS application areas such as capacity planning, accounting and chargeback, performance management, and system reliability analysis.
Underlying the CA MICS database is the CA MICS Platform. The CA MICS Platform enables the entire CA MICS system to work together. It consists of four interrelated facilities:

- Access and retrieval facilities allow applications to access, analyze, and use information contained in the CA MICS database.
- Data management facilities provide the data management functions for the CA MICS database and administrative functions for the system.
- Development and extension facilities provide tools for expanding and customizing the system.
- Reference facilities enhance expertise through in-depth documentation with online and printed access.

One of the major strengths of CA MICS is its ability to integrate data from diverse sources into a single CA MICS database. Since the data processed by CA MICS Accounting and Chargeback Option is taken from a unified CA MICS database, you can combine the necessary data with any other data that CA MICS stores regardless of its original source.

---

**Figure 1.1. The CA MICS IS Management Support System**
OTHER GUIDES YOU NEED

If you are responsible for setting product parameters and running CA MICS Accounting and Chargeback Option, or just want to learn more about using the product, look at the CA MICS Accounting and Chargeback Option User Guide. You can view it in CA MICS Document Access.

If you want to learn more about Enterprise IS Financial Reporting after reading Section 1.1 of this guide, see the Getting Started with Enterprise IS Financial Reporting Guide and Chapter 6 of the CA MICS Accounting and Chargeback Option User Guide.

If you are installing CA MICS Accounting and Chargeback Option in a CA MICS complex or unit, see the PIOM. Chapter 2 in the CA MICS Accounting and Chargeback Option User Guide directs you to the appropriate place in the PIOM. The PIOM is available in CA MICS Document Access or ask your CA MICS administrator.

This section contains the following topics:

1.1 Key Concepts (see page 10)
1.2 Major Features (see page 14)
1.1 Key Concepts

This section explains important concepts employed by CA MICS Accounting and Chargeback.

COST CENTER

A cost center is an entity in your organizational hierarchy such as Division, Department, or User. CA MICS Accounting and Chargeback supports up to nine levels of cost centers in variables called COSTCTR1-COSTCTR9.

The cost center levels form a hierarchy; for example:

```
  Division Sales                  COSTCTR1
       |                           |
       |                           |
       +--------------------------+
           |                     |
           |                     |
           +-------------------+
             |       |       |
             |       |       |
             +-------+-+-------+
                |       |
                |       |
                User 1 User 2 User 3 COSTCTR3
```

You tell CA MICS Accounting and Chargeback how many levels of cost centers you want to keep and what each level contains. This is called defining the cost center structure, or defining the cost center hierarchy. You also tell CA MICS Accounting and Chargeback how to populate the COSTCTR variables.

Each record in a CA MICS Accounting and Chargeback file contains COSTCTR1-COSTCTRn variables, where n is the number of levels that you defined. The COSTCTR variables are summarization keys in the files, and are used to identify the work being charged.

Because they are summarization keys, you can also use them to carry any information useful for reporting, such as project or application.
A charging element is a quantifiable item to which a rate can be assigned to create a charge. Charging elements may be resource (usage) based, such as CPU seconds or print lines; they may be transaction (business unit) based, such as number of deposits or number of checks; or they may be service (subscription) based, such as email or backup services.

CA MICS Accounting and Chargeback identifies each charging element with a four-byte key called a computation code. The associated computation code is stored in the CA MICS Accounting and Chargeback files instead of the name of the charging element.

Examples of charging elements and computation codes are:

<table>
<thead>
<tr>
<th>Charging Element</th>
<th>Computation Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Seconds</td>
<td>0022</td>
</tr>
<tr>
<td>Print Lines (in thousands)</td>
<td>0222</td>
</tr>
<tr>
<td>Checks Written</td>
<td>8001</td>
</tr>
<tr>
<td>Email Service</td>
<td>8351</td>
</tr>
</tbody>
</table>

Charging element and computation code are often used as synonyms within CA MICS Accounting and Chargeback.

A table of standard computation codes, the Charging Element Table, comes with CA MICS Accounting and Chargeback. You can add new computation codes to this table if you need one that it does not contain. Only those computation codes that you specify in another table, called the Rate Table, are actually used for charging.

A load center identifies items that have a cost associated with them, such as central processors, printers, disks, workstations, software, networks, environmental systems, or floor space.

When rates are being developed, load centers are identified and charging elements and rates are assigned to recover their costs. For example, CPU seconds may be charged to recover the cost of the central processor and a certain percent of
floor space and environmental costs.

CA MICS Accounting and Chargeback identifies each load center with a four-byte key called an invoice category. The associated invoice category is stored in the CA MICS Accounting and Chargeback files instead of the name of the load center.

Each computation code is associated with a single invoice category. So each invoice category identifies a group of computation codes that can be charged to recover the load center's costs.

Examples of load centers and invoice categories are:

<table>
<thead>
<tr>
<th>Load Center</th>
<th>Invoice Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Central Processors</td>
<td>1100</td>
</tr>
<tr>
<td>Control Units</td>
<td>1300</td>
</tr>
<tr>
<td>Personal Computers</td>
<td>1650</td>
</tr>
<tr>
<td>LAN Equipment</td>
<td>2400</td>
</tr>
<tr>
<td>Printers</td>
<td>3100</td>
</tr>
</tbody>
</table>

A table of standard invoice categories, the Invoice Category Table, comes with CA MICS Accounting and Chargeback. You can add new invoice categories to this table if you need one that it does not contain.

JOURNAL FILE

Journal files contain information on work units charged. A separate journal file is kept for each data source and type of work charged (for example, batch jobs, TSO sessions, CICS sessions, UNIX processes).

Each journal file record represents a single unit of work and contains the COSTCTR variables, SYSID, work unit identifier, billing timestamp, and a section for each charging element used to charge the unit of work.

Each charging element section contains invoice category, computation code, quantity, rate, and charge.

Journal files are created during the DAILY processing for the CA MICS Data Integration Application from which they are derived.
LEDGER FILE

Ledger files contain information on each charging element used. A separate ledger file is kept in the DAYS, MONTHS, and YEARS timespans of each CA MICS unit in which CA MICS Accounting and Chargeback is installed.

Each ledger file record represents a single charging element section in the journal file and is summarized by the COSTCTR variables, SYSID, invoice category, computation code, rate, and DAY, MONTH, or YEAR.

The ledger files are created and/or updated by the CA MICS Accounting and Chargeback steps in the DAILY, MONTHLY, and YEARLY jobs.

FINANCIAL RECAP FILE

The Financial Recap file contains summarized information on all charges, debits, credits, discounts, surcharges, and budgets in CA MICS Accounting and Chargeback. A single Financial Recap file is created each month and a year-to-date Financial Recap file is updated.

Invoices are produced from the Financial Recap file and you can export it to a general ledger or other financial system.

Each Financial Recap file record is similar in structure to a ledger file record and is summarized by the record type (debit, credit, etc.), COSTCTR variables, SYSID, invoice category, computation code, and rate.
1.2 Major Features

CA MICS Accounting and Chargeback Option is accessed through the CA MICS Workstation Facility (MWF). The MWF online facility allows you interactive access to all of the retrieval, processing, installation, and maintenance processes of CA MICS Accounting and Chargeback Option. You can use the online panels to run a report, change an installation parameter, or enter data.

CA MICS Accounting and Chargeback Option uses the SAS language for its processing functions. CA MICS Accounting and Chargeback Option generates all of the SAS code and JCL it requires to perform standard processing functions so that the only SAS routines you need to develop are those needed in exits and for special processing.

The major features of CA MICS Accounting and Chargeback Option can be grouped into the following categories:

- Enterprise IS financial reporting
- Standard accounting methodology support
- Pricing strategies support
- Rate simulation
- Comprehensive processing of any accounting data
- Interface capabilities
- Budgets
- General product flexibility

ENTERPRISE IS FINANCIAL REPORTING

Enterprise IS financial reporting is a comprehensive framework to help you analyze how corporate investments in IS technology are being used. It provides information from a variety of perspectives to address the needs of various members of the enterprise, including enterprise management, IS management, IS user management, and accounting administrators.

The information is organized in five management impact areas:

- Invoicing
- Client Support
- IS Financial Management
- Enterprise IS Financial Management
- Accounting Administration

Enterprise IS financial reporting provides information
as a mainframe report, such as an invoice.

Invoicing

Because invoices provide a summary of charges by user, they are an important source of information whether or not any actual revenue is generated. CA MICS Accounting and Chargeback Option provides a number of features, listed below, to give you flexibility in this area.

Each of these features can be specified differently for each cost center invoiced. This allows you to individually tailor your invoices to meet different requirements for different cost centers.

- **Variable Cost Center Level of Invoices**

  You can produce invoices at any cost center level, and invoices for different parts of your organization can be produced at different cost center levels. For example, you might elect to produce most invoices at the division (COSTCTR1) level, but produce invoices for a particular division at the department (COSTCTR2) level.

- **Flexible Line Item Formats and Subtotals**

  You can choose the content and format of the line items on each invoice. The line items can be either computation codes, invoice categories, or cost center values, and can show year-to-date and/or budget amounts as well as current month charges.

  A computation code line item format prints a single invoice line for each computation code and rate combination. It can show the computation code, its description, quantity, rate, charge, and year-to-date quantity and charge.

  An invoice category line item format prints a single line for each invoice category, and can show the invoice category, its description, charge, budget, and variance.

  A cost center line item format prints a single line for each value of a designated cost center level (such as project), and can show the cost center value, its description, charge, year-to-date charge, budget, and variance.

  You can choose optional line item subtotals for each of
1.2 Major Features

these line item formats.

- Supplemental Reports

  Supplemental reports can be produced after each invoice; you can choose either Detail or Rollup supplemental reports.

  - Detail supplemental reports print one page for each user who incurred charges.

  - Rollup supplemental reports print one page for each node in the cost center hierarchy below the level invoiced. For example, if the invoice level is division (COSTCTR1), you could get one supplemental report page for each department (COSTCTR2), and one page for each project in each department (COSTCTR3).

- Budget and Year-to-Date Information

  You can choose to print summarized budget and/or year-to-date information at the bottom of an invoice instead of printing it at the line item level.

  If budget information is shown at either the line item or summary level, a warning message is printed at the bottom of each invoice if the total charges exceed the total budget by an amount that you specify. This amount is called the fuzz value.

- Footnotes

  You can specify a one- or two-line footnote to print at the bottom of an invoice.

- Language Translation

  You can create translations for invoice category and computation code descriptions and show them on invoices instead of the standard English language descriptions.

- Billing Status (BILL or NOBILL)

  You can specify the billing status (BILL or NOBILL) for each cost center that is invoiced.

  You can produce invoices for all billable cost centers, all non-billable cost centers, or for both.
o Personalized Invoices

You can specify a name and address for each cost center and it will print on the appropriate invoice or supplemental report.

o User-Written Invoice Routines

If none of the standard invoice formats meet your needs exactly, you can write your own invoice routines. Different routines can be used for different cost centers.

If your format is close to one of the standard formats, you can model your routine from one of the standard routines and thus retain support for the standard options.

STANDARD ACCOUNTING METHODOLOGY SUPPORT

CA MICS Accounting and Chargeback Option supports two primary accounting methods, transaction accounting, and resource accounting.

o Transaction accounting, sometimes referred to as "functional accounting," charges a constant rate per execution of a specific item of work (for example, $0.14 per CICS subscription update transaction). Transaction accounting is often the preferred method of charging for interactive operations such as CICS, IDMS, IMS, and TSO.

o Resource accounting uses the value of the charging elements multiplied by the applicable rate (for example, the number of CPU seconds multiplied by the charge per CPU second) to provide a single charge for each resource used.

You can use either, or both, of the primary accounting methods for allocating charges to users of data processing services. Typically, if you use transaction accounting to charge for interactive system usage, you will use resource accounting to charge for batch, DASD, and network usage.

Within each accounting method, CA MICS Accounting and Chargeback Option supports differential charging and various other pricing strategies.

PRICING STRATEGIES SUPPORT
Pricing strategies that you can employ with this system include the following:

- Differential charging
- Control variable charging
- Standard rates
- Minimum charging for a unit of work
- Zero-balance accounting
- Prorating
- Discounts and Surcharges

Each of the major pricing strategies that CA MICS Accounting and Chargeback Option supports is discussed in detail below.

**Differential Charging**

Differential charges are adjustments to the total charges for a unit of work that are applied after all resource consumption factors have been calculated. Differential charges are computed by multiplying a factor (discount or premium) and the charges accumulated for a unit of work.

Differential charges include priority differentials and zone (shift) differentials. Priority differentials are applied before zone differentials.

- **Priority Differentials**—CA MICS Accounting and Chargeback Option supports priority differentials for batch, SNA network, and VSE/POWER work. To activate priority differentials, you define a factor for each service level to increase or decrease basic charges for the level of service received. For example, a user who requested and received a Priority 1 job within the targeted time of 10 minutes might be charged two times the basic fee for the express service. However, if that user request was not fulfilled for three hours, the user would be charged a lower rate based on the service actually received.

- **Zone (Shift) Differentials**—You can apply differential zone accounting factors to all work. To apply shift differentials, you define a multiplier that increases or decreases basic charges for a specified shift or service period (zone). For example, users who perform work during the third shift or on weekends might receive a 50% discount for using the less busy hours.

**Control Variable Charging**
Control variable charging allows you to assign different rates to a charging element based on the value of a control variable. For example, if the charging element is the number of print lines and the control variable is the kind of form on which they are printed, then the rate that you charge to print a certain number of lines would vary depending on the form on which the lines are printed.

Standard Rates

A standard rate is a specific price for a product or service over an extended period of time such as a year. The rate stability offered by standard rate pricing allows users to better predict expenditures for budget and planning purposes.

Minimum Charging

Minimum charging is a set minimum fee for a unit of work. The charge for the actual work may be greater than the minimum amount, but it will not be lower.

Zero-balance Accounting

Zero-balance accounting, also known as variable rate charging, is the process by which a data processing operation recovers its costs by distributing those costs among the users of the computer services during the accounting period.

The CA MICS Accounting and Chargeback Option approach to zero balancing uses standard rates throughout the period so that the approximate charges incurred by users are known. You may perform special processing at the end of an accounting period to create adjustment credit or debit entries to conform to a zero-balance charging process.

Prorating

The prorating process distributes the charges for a unit of work according to a user-defined set of percentages. For example, when a batch job is run for the benefit of more than one cost center (user) and you use prorating, each user is charged a prorated percentage of the charges incurred to run the batch job.
With CA MICS Accounting and Chargeback Option, you have the option of either prorating charges when they are originally computed (journal file prorating) or prorating charges at the end of the month, after they have been summarized (ledger file prorating). You can also perform prorating on external files.

In addition, you can use actual historical data to do "what if" analyses with the system's rate simulation facilities.

Discounts and Surcharges

Surcharges allow you to add a fixed percentage to an invoiced amount. They can be computed at the computation code, invoice category, or cost center level. You can specify different surcharges for each user invoiced.

A discount is a deduction from the cost center's charges. Three types of discounts are available:

- Volume discounts where the discount percentage is associated with a charge threshold. For example, a 10% discount for amounts over $10,000 and a 20% discount for amounts over $100,000.

- Invoice Category discounts are specified as fixed percents by invoice category and applied to the total charges for the invoice category on an invoice.

- Cost center discounts, either lump sum or fixed percentage, are applied to the total charges on an invoice.

You can specify one or more of these discount options for each user invoiced.

RATE SIMULATION

The rate simulation facility of the system applies hypothetical new rates to detail data from the audit files to produce reports that show the effects of the rate changes on revenue areas, cost centers, and charging elements.

COMPREHENSIVE PROCESSING OF ACCOUNTING DATA

CA MICS Accounting and Chargeback Option processes data from CA MICS files, CA MICS Field Developed Application files,
external files, and manually entered data. Any type of accounting data is eligible for processing.

- CA MICS Files--You can charge for usage based on standard CA MICS elements or on any combination of data elements in the CA MICS database (for additional details, see Interface Capabilities, below).

- CA MICS Field Developed Applications--Any CA MICS Field Developed Application can provide data for charging. The CA MICS Field Developed Application Directory, which is distributed with the CA MICS Product Support Program, lists many CA MICS Field Developed Applications along with user contact information.

- External Files--CA MICS Accounting and Chargeback Option can also charge for usage that any external management system has recorded. For example, if you use personnel or payroll processing measurement systems, you can charge users for transactions within those systems.

- You can enter standard or recurring debit or credit adjustments to charges. You can also batch load these adjustments.

  Debits are summed with corresponding charges (those having the same cost center values, invoice category, computation code, and rate) each month and appear as line items on the invoices.

  Credits can be subtracted from the corresponding charges, so that the net amount is shown on the line item, or they can be summarized and shown on a single line at the bottom of invoices.

INTERFACE CAPABILITIES

CA MICS Accounting and Chargeback Option can use data from the CA sources listed below. These products allow both transaction and resource accounting via a standard interface to CA MICS Accounting and Chargeback Option. Their files contain data in a form that CA MICS Accounting and Chargeback Option can use to bill for the use of system resources.

- I/S Inventory and Assets Manager
- CA MICS Analyzer for HP Performance Collection Software
- CA MICS CICS Analyzer
- CA MICS Space Analyzer Option
12 Major Features

- CA MICS DB2 Analyzer
- CA MICS IDMS Analyzer
- CA MICS IMS Analyzer
- CA MICS Batch and Operations Analyzer
- CA MICS SNA Network Analyzer
- CA MICS Analyzer for TSO
- CA MICS VAX/VMS Analyzer
- CA MICS VM and CMS Analyzer
- CA MICS VSE/POWER Analyzer
- CA MICS WEB Analyzer

Several hundred standard charging elements are distributed with CA MICS Accounting and Chargeback Option. However, the system can use any set of elements from the CA MICS database, and you can add elements for your Field Developed Applications and external files.

BUDGETS

CA MICS Accounting and Chargeback Option allows you to enter budgets at the cost center, invoice category, and computation code levels. Computation code budgets can be expressed in resource units (like CPU seconds) as well as in monetary amounts. These budgets can be compared to charges at the corresponding level on invoices.

Note that the total budget for an invoice category is the sum of the invoice category budget and all associated computation code budgets. Similarly, the total cost center budget is the sum of the cost center budget and all invoice category and computation code budgets for the cost center.

GENERAL PRODUCT FLEXIBILITY

CA MICS Accounting and Chargeback Option also provides several flexibility features:

- A user-defined cost center structure provides the ability to organize, summarize, and report charges by the major groups in your organization. A hierarchical structure allows multi-level reporting.

- An extensive series of panels make setup and modification easier. Detailed help is available for every panel.

- Charging strategies are provided for selected resources. These strategies discuss the issues involved and make
recommendations on charging elements and methods.

- A Data Dictionary describes the information found in the CA MICS Accounting and Chargeback Option files and how that information was derived.

- An alternate calendar is an optional capability that allows variations from the standard calendar to support an alternate fiscal year or different financial months. CA MICS Accounting and Chargeback Option files can use one calendar, while the other CA MICS database files use another calendar.

- Rate effective dates allow you to input planned rate changes in advance so that they will take effect automatically for work processed on or after the specified effective date.

- Multi-site accounting allows installations that have data centers at more than one physical site to transmit CA MICS Accounting and Chargeback Option files from remote sites to the central site and produce a consolidated invoice.

- You can modify almost any feature, including reports, algorithms, and processes, to fit your needs. The defaults and samples that are distributed with the system provide invoice categories, credit categories, charging elements, and charging algorithms that you can modify. In addition, you can customize the title and billing address information on any report.

- A wide range of standard exits allows you to tailor the product to the specific needs of your enterprise.

- You can export data to other financial systems; this gives you more ways to use your financial information.
Chapter 2: COST ACCOUNTING CONCEPTS FOR IS ORGANIZATIONS

An information system (IS) accounting function should:

- Record the consumption of data processing resources at the most basic level of usage (for example, job and session).
- Classify the usage both by user (cost center) and by resource consumed (load center).
- Summarize the information based on the classifications that will impart the greatest amount of information to those who use it without obscuring important facts.
- Report the summarized information at regular intervals in a manner most likely to reflect significant trends and characteristics.

According to Welsch, Zlatkovich, and White (page 32 of Intermediate Accounting, published by Richard D. Irwin, Inc.) in accounting terminology the process of recording the detail data is called "journalizing," after the journal in which it is recorded. The classification process is called "posting" the information to the appropriate account. The summarized record of the accounts into which a transaction is posted is called the ledger.

In large accounting systems, multiple ledgers frequently keep only one type of account. Thus, the ledger itself may well serve the reporting purpose. In that case, a summary ledger, the general ledger containing all accounts, is also maintained. The periodic process of publishing the contents of the general ledger for the organization and other interested parties is known as producing the Statement of Financial Position.
Few data center accounting systems will ever be called upon to produce a Statement of Financial Position, although they may contribute to the overall corporate statement. However, data processing managers need to allocate some portion of their costs to those who actually consume the resources. Thus, a computer accounting system can approximate the requirements of a business accounting system.

NOTE: Some profit-oriented organizations prefer to apportion not the costs incurred, but some percentage of the actual monetary value of the processing performed.

Allocating dollars to the users (called chargeback) is dependent on two processes: measuring the resources consumed by the user and determining a dollar representation of those resources. The remainder of this chapter discusses the first of these two processes, measuring the resources consumed, or computer accounting. The second process, pricing, is discussed in Chapter 3.

It may seem easy to determine the quantity of resources consumed by a user, since, unlike other equipment, the computer will do it for you. Unfortunately, it is often difficult to make a distinction between a resource and a product, or what was consumed and what was produced. The following example by Melvin J. Strauss (page 45 of the book Computer Capacity, A Production Control Approach) shows that the relationship between some of the accepted measures of computer utilization and their cost is even more confusing:

XYZ Corporation purchased a bank of disk drives for $623,000. XYZ's chargeback algorithm recovers for disk utilization by charging for EXCPs. If XYZ had purchased a box of EXCPs, its cost would have been a certain number of cents for each consumable EXCP. Everything from the vendor's brochure to the standard lease or purchase agreement would be stated in terms of EXCPs. However, XYZ did not buy EXCPs; it bought disk drives, and there is no rigorous method of determining how many EXCPs the disk drives might contain.

The remainder of this chapter describes the characteristics of a computer accounting system, discusses some of the popular methods of accounting for data processing utilization, and points out some pitfalls.

This section contains the following topics:
2.1 Essential Characteristics of an IS Accounting (see page 28)
2.2 Budgeting Concepts (see page 31)
2.3 Resource Accounting (see page 40)
2.4 Transaction Accounting (see page 47)
2.5 Differential Charging (see page 62)
2.6 Additional Accounting Options (see page 66)
2.7 DASD/DFHSM/SMS Storage Accounting (see page 70)
2.8 Client/Server Accounting (see page 88)
2.9 UNIX Accounting (see page 96)
2.10 DB2 Accounting (see page 103)
2.11 Network Accounting (see page 112)
2.1 Essential Characteristics of an IS Accounting

Experience has shown that the three most important characteristics of an accounting system are consistency in charging, equity in the cost distribution method, and simplicity of approach.

CONSISTENCY IN CHARGING

Consistency in charging means that a unit of work processed with the same data activity has a consistently repeatable charge that does not vary because of the multiprogramming environment. This characteristic is by far the most important feature of a computer accounting system. The two factors that must be consistent are resource measurement and resource pricing.

Consistent resource measurement can be achieved to an acceptable degree only if multiprogramming, multiprocessor, and paging factors are taken into consideration. More importantly, such factors as elapsed time cannot be used as a charging base because the amount of time a job resides in the system cannot be measured with consistent or repeatable results.

Consistency in resource pricing presents a completely different problem. Consistency in resource pricing occurs when the charge for a resource such as CPU time remains the same throughout the year and does not vary on a monthly basis because of a fluctuation in data center utilization.

This pricing approach is a radical accounting departure for many organizations. Traditionally, most organizations have operated on a break-even policy, using the month as the accounting cycle. This accounting policy is unacceptable for users, because they cannot analyze past monthly expenses against one another or project future costs. The accounting dilemma that is a result of this zero-break-even policy is illustrated by the following example:

The accounting department of company XYZ was charged $25,000 for its use of the firm's computer system in the month of January. For the month of February, the accounting department projected at least a $5,000 decrease from the January costs because its workload would decrease by at least 20%.
However, the firm's computing center used a monthly zero-break-even policy. In January, the accounting department's usage represented 10% of the computing center's workload for that month. Since the center's total costs for January were $250,000, the accounting department was billed 10% of that amount, or $25,000.

In February, however, even though the accounting department's usage of the center decreased, so did everyone else's usage. For February, the center's total costs were again $250,000. The accounting department's usage represented 20% of ALL computer center usage for February, so the bill issued to the accounting department was $50,000, instead of the $20,000 they expected.

In order to have an accounting system that approaches user acceptance, the monthly break-even policy must be eliminated and standard rates instituted for extended periods of time.

EQUITABLE COST DISTRIBUTION

Due to industry concepts such as economy of scale, multiprogramming, and timesharing services, users have become aware of the need for equitable charging. Equity in computer accounting means charging users for exactly the resources used by each unit of work, no more and no less. Additionally, accounting systems should not directly charge users for any system cost that the users cannot control.

An inequitable accounting system causes significant repercussions. Consider a system that is charging for jobs according to the amount of CPU time that the job has used. Observe two different jobs and how they consume resources:

- JOB A is CPU-bound and uses 200K of memory, thirty minutes of CPU time, and one printer.
- JOB B is both CPU and I/O bound and uses 700K of memory, thirty minutes of CPU time, and three tape drives.

Both Job A and Job B used thirty minutes of CPU time. Based on the CPU charge concept, these jobs are charged exactly the same amount, yet they consumed totally different levels of system resources.
In this example, it is obvious that an inequity exists in resource measurement, but, like the problems of consistency, equity is a function of both resource measurement and resource pricing. Equitable resource measurement quantifies how a job has used each available system resource. Equitable pricing ensures that the charge for resources is not disproportionate to the cost. An example of inequitable pricing is a case where resources such as tapes, direct access, storage, and CPU time are charged out, but the CPU time has a disproportionately higher price than the other three resources when their actual costs are considered. Under this accounting policy, the CPU-bound job is charged more than its fair share, while the I/O bound job is charged less than its fair share.

SIMPPLICITY OF APPROACH

A system that ensures consistent and equitable resource costing is inevitably complex in the required operating system interface and in the accounting process itself. However, users do not have to be aware of the complexity. Instead, to encourage user acceptance, the accounting system should be packaged simply.

All too often, accounting systems for computer usage have been plagued by packaging deficiencies. The terminology of the operating system is often used in the chargeback statement to the user (for example, $1.35 per 1000 EXCPs). Or, instead of using stated rates, many operating system accounting routines have used complex algebraic algorithms.

To satisfy the typical business structure, algebraic formulas should not be used to express an accounting function. Algebraic expressions or complex mathematical relationships often hinder user understanding and acceptance.

If a meaningful accounting system is desired, you must consider three objectives: consistency, equity, and simplicity. An effective system ensures that the requirements of both management and personnel are met.
2.2 Budgeting Concepts

Most organizations implement accounting and chargeback either to improve their IS resource allocation and control or to bolster their IS management and planning. Having gained insight into the actual period-to-period IS expenditures, many find that they need a base of comparison, independent of the actual expenditures, to judge the effectiveness of their plans and controls. Most choose to make that comparison to an individual user’s IS budget.

From a resource control perspective, the budget represents an approved maximum allocation expressed in monetary, rather than technical, terms. From a management planning perspective, the budget quantifies anticipated expenditures for which sufficient resources have to be made available.

The potential for a serious budget development, analysis, and approval effort to reduce actual expenditures should not be underestimated. After all, by the time an actual expenditure is reported by the accounting system, it has already been incurred and, in many cases, represents an ongoing cost for the user into the foreseeable future. The beauty of working with budgets is that the costs they represent have yet to be incurred.

The following sections provide an introduction to budgeting philosophies, budgeting development, and utilizing budget information in reporting scenarios:

1 - What is a User Budget?
2 - Budget Detail, How Much is Enough?
3 - Steps in Developing a Budget
4 - Reporting Budget Information
2.2 Budgeting Concepts

2.2.1 What is a User Budget?

Most IS organizations today serve a number of largely independent clients. These clients may be organizational units within the IS organization's own parent enterprise or they may be entirely separate entities. Regardless of the relationship, many of these clients feel that IS is better equipped to assess and project their IS usage than they are. This dual perception of IS usage being difficult to predict, and the IS organization possessing greater insight, has resulted in IS providing substantial input into, and subsequent reporting of, user budgets. This entire practice is called User Budget Management.

A user budget is a formal expression of an individual cost center's expected IS usage, stated in financial terms. It is closely related to the IS organization's capacity plan except that it encompasses only billable usage, is subdivided by cost center, and is enumerated in monetary, rather than resource, units.

The definition of a budget implies that some planning was involved in its development. However, in the case of an IS user budget, relatively little planning may actually have been done by the user, who is therefore often unwilling to accept much responsibility if the expectation proves erroneous. This can cause problems for IS, who finds itself not simply a supplier of services, but also limited by an implied sales "cap."

Ultimately, the size of a particular user's budget depends upon many things. Assuming that management allows the budget to be based solely on usage projections (rather than arbitrarily limiting it to some level), it will vary directly with the anticipated volume of work. However, the actual expenditures may vary because the unit cost of the work, when it is performed, differs from that assumed by the projection.
Factors which most often cause variances from budget projections are unanticipated changes in "technical" resource consumption. For resource-based accounting methodologies, these changes may occur in any billable resource. For example, the amount of CPU time required to process a given transaction or the distribution of data across various levels of storage (online, offline, archived, etc.) may change. Transaction-based accounting methodologies are less susceptible to these types of variances, however the transaction mix for a particular system, user, etc. may change, causing similar budget variances.

Budget variances, however, are the motivation for all budget-oriented planning and control. Identifying where a variance exists and taking steps to eliminate it, either from the production process or the budget model, is the essence of financial management. Having sufficiently detailed budget information to identify the cause of variances is discussed in the following section.
2.2.2 Budget Detail, How Much is Enough?

Some organizations feel they can plan adequately on the basis of an individual IS user's total budget. These organizations typically set objectives to limit budget growth or to reduce expenditures on an across-the-board, fixed percentage, basis. However, there is a groundswell of functional business managers who are demanding that budget projections be broken down by user, application, and specific resource category. These managers feel they need detailed unit costs upon which to base their budget decisions.

CA MICS Accounting and Chargeback supports budget information at three levels of detail: Cost Center, Invoice Category, and Computation Code. All budget entries apply to a particular cost center value, however invoice category and computation code budgets allow further detail within a cost center level. Budget data can be entered at any or all of the three levels, however, all budget data will be summarized to the cost center level being reported upon by the invoice and other standard reports. Computation code budgets can be entered in resource units (for example, print lines) as well as in money amounts.

The degree of flexibility provided by CA MICS Accounting and Chargeback to classify data elements (cost centers, invoice categories, and computation codes) precludes identifying definite budgeting levels and strategies. However, some scenarios can be suggested.

In general, budgeting at more than one cost center level helps to increase management control of utilization by providing objectives for the managers who directly supervise the work which is responsible for the charges. This control can be further increased if an installation utilizes some levels of the cost center structure to identify non-organizational groupings such as application system, test or production status, etc. The purpose of such classifications is to provide more detail about IS consumption within organizational units with diverse IS usage.

Similarly, budgets at the invoice category or computation code level can serve to identify specific resources which are being used to excess. This can be extremely helpful in identifying the causes of a total budget overage.
2.2.3 Steps in Developing a Budget

Once the enterprise has determined the level of detail to incorporate into the budget, the actual budget development process can be broken down into four steps:

- Initial forecast development
- Forecast update and validation
- Consolidation and management reporting
- Finalize

The initial forecast development is normally based on unit resource volumes projected from historical accounting information. The projected volumes are then multiplied by the applicable rates to compute anticipated expenditures. The expenditures are then summarized and subjected to a "reasonableness check" usually by comparison with the previous budget. While this methodology is basically sound, it harbors two potential pitfalls. First, extremely large projected growth rates seldom pass the reasonableness test even if no evident source of error can be found. Second, even if the projected expenditures are accurate in total, they may vary substantially from the month-to-month actual costs.

Poor correlation between the projections and the actual expenditures often results from an ineffective forecasting technique or model. Many organizations simply do a linear projection of the billable elements over the planning horizon. Unfortunately, IS usage is rarely linear. In most cases studied, there were periodic spikes in usage usually tied to a seasonal fluctuation in the business. Consequently, in nearly every month, there was a variance from the budget that the users were required to explain. In the rare cases where the users are not concerned about the variance, a linear forecast will work, but, for most organizations, a time series analysis (which will incorporate the seasonal fluctuations into the projection) is more appropriate.

Now comes the most critical step in developing a realistic budget--validating and updating the forecast. The initial forecasts should be displayed on easily read reports which incorporate plenty of white space for users to write in changes. Many enterprises download the forecasts into a spreadsheet because they are easy to manipulate and they yield machine readable output that can be used to quickly update the initial forecasts.
2.2 Budgeting Concepts

The mark-up reports or spreadsheets are then used to manually factor in plans and changes which were not reflected in the historical accounting data. These changes can be caused by a number of factors. Some of the possibilities include:

- Changes in business processes
- Changes in business volumes
- New applications
- Enhancements to existing applications
- Completion of system implementations
- Introduction of new technologies (hardware or software)
- Implementing an information center
- New IS service offerings or changes in the current offerings

This step involves communicating (usually in meetings) with end users, business planners, IS capacity planners, and application development staffs. Once the anticipated changes have been incorporated into the plan you are ready for consolidation and management reporting and iterative tuning of the plans.

The consolidation process is simply updating the master copy of the forecasts with all of the information collected in the interview process. This is a good point for a sanity check that compares the consolidated forecasts of units with the IS capacity plans to determine if the users forecasts of demand bear any resemblance to the IS organizations ability to service them. If there is a significant discrepancy, work with the capacity planners to determine which one is correct. The two forecasts are frequently done independently and can yield very different results. Resolution of the difference may involve going back to some of the original interviewees. Once this is resolved you can test the rates by comparing the user budget projections against the IS spending plans.

Again, these may be very different for many reasons. If your objective is full cost recovery, a rate adjustment may be required to balance the user billing with IS spending plans. There is also a possibility that you will have identified an unexpected demand that will result in a change in the IS spending plans to satisfy the requirement.
Now you're ready for management reporting, both internal and external to IS. The IS management team wants to understand how their services will be consumed and by whom and user management team will be interested in how the forecast fits within their spending plans and business objectives. During this phase you should expect to adjust the plans several times. Today's emphasis on cost reduction will prompt questions such as "How can we get more out of IS at less cost?" and "Which programs can we eliminate to get IS costs within our spending plans?".

Many organizations will begin to take shortcuts at this point. These generally occur when upper management edicts demand a flat percentage reduction of IS costs and the forecasts are simply reduced by that percentage. Keep in mind that your forecast represents the plans of the business units. They must be involved in and committed to any cost reduction activity. Costs don't usually go down because someone thinks they are too high, some action plan must be devised and implemented to make it happen. We know of organizations in which the credibility of IS is negligible because of a flat reduction that resulted in users being significantly over budget after the first month of the fiscal year.

Organizations that have not attempted to rigorously develop IS budgets in the past should not attempt to apply this process to the entire IS workload at one time. Start with the largest workloads and work with them until the model is refined and reasonably accurate. This approach should represent a high percentage of the budget and is often sufficient. To arrive at a total budget, group the smaller workloads into a single category that you address as a whole. It is usually not worth the effort required to address everything in detail. Nor is it likely that you will every be able to model everything accurately at a reasonable cost. However, the point of diminishing returns will vary from organization to organization.
2.2 Budgeting Concepts

2.2.4 Reporting Budget Information

The purpose of a budget is to provide a criteria with which the relative "goodness" or "badness" of a particular level of consumption can be assessed. Providing this comparison is the function of budget reporting.

The most basic and generally applicable form of budget reporting is the comparison of actual charges to budgeted amounts. The budget-versus-actual comparison implies a good/bad criteria with relatively little requirement for interpretation and no "shades" of meaning. However, the interpretation still must be based on an understanding of the management objectives. For example, an actual expenditure significantly under budget is likely to be perceived as positive to an enterprise which is attempting to reduce expenditures to a minimum. However, the same comparison may be unfavorable in an enterprise which is attempting to expand rapidly to meet market demand.

Similarly, when attempting to address budget variances, the magnitude of the budget as well as the percentage variance need to be considered. A one-hundred percent variance of an item budgeted for $500 is less significant to the enterprise than a one percent variance of a $1,000,000 budget. CA MICS Accounting and Chargeback provides a facility called the Budget Overrun Fuzz Value to limit the reporting of insignificant budget variances.

The use of historical budget reporting can add an extra dimension to simple budget-versus-actual comparisons. One concern of many IS organizations is to maximize their ability to proactively identify problems. Simple budget-versus-actual reporting only identifies expenditure problems after the fact. But, by trending the percentage of unused budget authority over time, a potential budget overage in the future can be anticipated and possibly avoided.
Historical analysis may also uncover errors in the forecasting model used to develop the budget. If an entity has a budget which fluctuates in magnitude significantly from period to period, an overage in one period may only indicate that the forecast does not correctly identify seasonal variances. By comparing summarized budget and actual values over several periods (for example, for a quarter), a single period variance can often be seen in perspective.

Handled well, the budget reporting effort will yield important management information, a clear understanding of demand for IS services and how IS costs are distributed within the enterprise.
In the early days of computing, when only one task at a time could run on a machine and any task used virtually all of the machine’s associated resources, the elapsed time to run the job was an adequate measure of utilization. This method charged a job according to the amount of time the job resided in the system. Today, with advances in technology, the elapsed time is a function of the multiprogramming mix and cannot be controlled by the user. The elapsed time method of charging provides inconsistent as well as inequitable results for everyone concerned. Almost every advance in technology has made it more difficult to determine who used what and when.

Numerous methods for determining resource utilization have been developed. The common factor among all of them is the attempt to approximate a measure equivalent to the classical elapsed time.

Resource accounting is the technical method of accounting for resource utilization. Usage is measured in terms of definable computer resources such as CPU time, I/O, and memory, as opposed to accounting based on work performed (for example, checks produced). For accounting purposes, the resources available to a unit of work fall into three categories: dynamic resources, dedicated resources, and miscellaneous resources.

**Dynamic Resources**

CPU time and channel time are dynamic resources. CPU time is the amount of time the task is in control of the CPU. Channel time is, at best, a measure of the time taken to transfer data from main storage to devices and back. These resources are charged out according to actual utilization. In other words, the system might charge out CPU time at $260 per hour of utilization and channel time at $32 per hour.
DEDICATED RESOURCES

Dedicated resources consist of main storage and all I/O devices. These are charged not according to utilization, but according to the time the user has been responsible for their allocation. For example, the true costs of a tape drive, once data transfer time to the device has been considered, is not the utilization time, but the time that it has been allocated and therefore unavailable for use by any other job.

MISCELLANEOUS RESOURCES

Miscellaneous resources are comprised of spooling costs and operator costs. These resources should be charged according to the number of times they occur. Spooling costs include print lines, as well as the use of special forms. Operation costs are items such as volume mounts and operator message replies. A typical charge for a job includes the charge for the number of print lines used to print the output and a charge for each volume mount required by the job.

The following sections outline different types of resource accounting methods:

1 - CPU Time
2 - CPU Time/Memory Utilization Accounting
3 - CPU Occupancy Time
4 - Transaction Residency Time
5 - Resource Occupancy Time
2.3 Resource Accounting

2.3.1 CPU Time

The first attempt in the industry to charge for a specific resource was to charge a unit of work solely for the amount of CPU time consumed. While CPU time is a valid charging resource, it cannot be used alone, because work uses other significant resources that must be considered. z/OS (paging) and multiprocessor systems (spin locks and so on) add more inequities to CPU time accounting systems.

z/OS distinguishes between two types of CPU time: Task Control Block (TCB) time and System Request Block (SRB) time. In theory, SRB time is the amount of CPU time consumed by the operating system on behalf of a task, while TCB time is the CPU time consumed by the task itself. Thus, SRB time is assumed to be variable, depending on the multiprogramming mix, while TCB time is assumed to be constant and therefore a good accounting measure. Unfortunately, numerous problems of both inclusion and exclusion, as well as system architecture, tend to corrupt the TCB time measurement.

With all its complexity, CPU time is still probably the most commonly used measurement of data processing resource consumption.

2.3.2 CPU Time/Memory Utilization Accounting

One method of charging for CPU usage is to consider CPU time and the amount of main storage used or allocated. In this method, the two resources are multiplied by one another to arrive at the chargeable resource of the Kcore second, Kcore minute, or Kcore hour. To illustrate this charge, consider the following example:

Job step A allocates 100K of main storage and uses 16 minutes of CPU time. The resource charge for this run is $.25 per Kcore minute. In this example the job step charge is $400.00 (100K * 16 minutes = 1600 Kcore minutes * $.25 per Kcore minute = $400.00).

One consideration for this method is that not all resources are accounted for. For example, a CPU-bound job may be charged proportionately too much than a job that is both CPU and I/O bound.
2.3.3 CPU Occupancy Time

CPU occupancy time, also known as pseudo-elapsed time, approximates how long a job would be in the system if it ran all by itself, without the influence of the multiprogramming mix. Two basic methods compute pseudo-elapsed time: the algorithmic and wait-time methods.

ALGORITHMIC APPROACH

The algorithmic approach uses the following type of formula to approximate the occupancy time:

\[
\text{Occupancy Time} = \text{CPU Time} + (dms \times \text{Number of I/Os})
\]

where "dms" is the average duration in milliseconds of one input/output (I/O) operation.

The error in this approach, however, is that the user is charged the full system rental, when the job may have required only minimal system resources. For example, suppose you submitted a tape copy program that would run for ten minutes if it was the only job in an environment with thirty tape drives and numerous disk drives. You would pay for much more than the job was consuming.

Another problem with the algorithmic approach is its inability to consider other important characteristics such as device speed, overlap, channel separation, and access methods. For instance, an indexed-sequential update operation and a sequential read operation would be charged exactly the same amount because they are one I/O operation each. In actuality, however, the index-sequential is much more time-consuming than the sequential operation.
WAIT-TIME APPROACH

The wait-time approach is based on the operating system measurement wait time. Wait time is calculated with the following formula:

\[
\text{Occupancy Time} = \text{CPU Time} + \text{Wait Time}
\]

where Wait Time is the amount of time that the task was voluntarily waiting for its own I/O function to complete.

In theory, the wait-time approach is the most accurate. However, measuring wait time accurately and equitably presents another problem. In most uses of this approach, wait time is not determined solely as the amount of time the task required, but includes factors such as access contention that is not the fault of the task.
2.3.4 Transaction Residency Time

Transaction residency time attempts to measure the time a transaction is active in the processor for interactive systems such as CICS and IMS. The diagram below helps explain this measurement:

When a message enters the system, it is transmitted to the CPU, where it resides until the processing completes. The period between the time of input and the time the task is attached is not included in transaction residency time, nor is the period between the time the task is detached from the CPU and the time the output message is received.

Because transaction residency time is dependent on the multiprogramming mix, it varies with the performance of the computer and the number of other transactions in the system. Since the same task does not always have the same transaction residency time, it is an inconsistent, and therefore unsatisfactory, method of measurement.
2.3.5 Resource Occupancy Time

One common approach to account for device usage is to charge for the utilization (I/O interrupts) of a device. This approach, although widely practiced, has some drawbacks. The charge for a device should consider the amount of time the device is allocated in addition to its utilization. For example, a job allocates three tape drives and uses only one of them. With utilization accounting, the job step would incur no charge at all for the two devices that were allocated but never used. As a result, only one part of the true cost is accounted.

Resource occupancy time applies pseudo-elapsed time to the allocated resources. It is in the accounting for dedicated resources that the use of occupancy times is truly valid. Because the user does not have control over the multiprogramming environment, it is not consistent or equitable to charge for a tape drive according to the actual elapsed time of the job step. However, it is valid to charge for that tape drive according to the pseudo-elapsed time or occupancy time, because the user does have control of that measurement. In other words, the dedicated resources can be charged out according to how long the job step would have used them in a single-job environment. This permits each resource to be examined according to its cost and allocation. The following example illustrates charging for dedicated resources.

Job Step A uses five tape drives. The measured times for this job step are:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Elapsed Time</td>
<td>60 Mins</td>
</tr>
<tr>
<td>Occupancy Time</td>
<td>30 Mins</td>
</tr>
<tr>
<td>CPU Time</td>
<td>11 Mins</td>
</tr>
</tbody>
</table>

The charges for this job step are:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Time</td>
<td>11 minutes @ $15.00/minute = $165.00</td>
</tr>
</tbody>
</table>

Dedicated Resources:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tape Drives</td>
<td>5 tapes allocated for 30 minutes each = 150 tape occupancy minutes (TOM) @ $0.10/TOM = $15.00</td>
</tr>
</tbody>
</table>

Total Step Cost $180.00
2.4 Transaction Accounting

An intermediate sized automobile sells for $10,000, an appendicitis operation costs $1500, an attorney drafts a will for $250, and a steak dinner sells for $14.95. Each of these products and services has been produced and analyzed by its management for a long enough period of time to establish "standard" charges per unit, product, or service. In fact, the customers demanded this form of standard unit charging so that they could effectively understand and plan their expenditures.

In only a few instances has this charging approach been successfully used in charging for computer services. Generalized computerized payroll systems charge their users according to the number of employees processed or checks generated. Computerized reservation systems charge their users a standard fee per reservation request. Computerized mail circulation systems charge per organization or person mailed. In each of these cases, we are dealing with relatively complex computer systems that have been able to implement a standard charge per unit product or service.

Transaction accounting, as used here, means the concept of charging for computer services by applying a standard charge per unit of input or output (transaction) processed, such as $2.50 per payroll check, $4.75 per reservation request, or $0.26 per addressed letter. The standard fee must remain constant for a long enough period of time, such as one year, to ensure accounting stability.

The idea of transaction accounting has always been well received by the users of computer services and by corporate management because it is such a simple concept to understand. Additionally, transaction accounting is the charge-out philosophy in most facets of business outside the computer services area.

The following sections discuss transaction accounting in more detail:

1 - Resource Accounting/Transaction Accounting Examples
2 - Implementing Transaction Accounting
3 - Summary of Transaction Accounting
2.4 Transaction Accounting

2.4.1 Resource Accounting/Transaction Accounting Examples

The examples below illustrate the difference between current resource accounting approaches and transaction accounting. They compare a traditional batch charge-out case to TSO transaction accounting examples.

BATCH SYSTEM EXAMPLE

A batch payroll system consists of numerous program processes that are directed at updating the master payroll file and producing payroll checks and associated accounting reports such as the check register report. The data center most likely accumulates charges either at the job or job step level based on the programs executed and the computing resources used, in accordance with some of the charging methods previously discussed. The actual charges to the user are expressed in terms the data center is able to measure, such as:

<table>
<thead>
<tr>
<th>JOBNAME</th>
<th>PROGRAM</th>
<th>TOTAL</th>
<th>CPU</th>
<th>I/O</th>
<th>MEMORY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CHARGE</td>
<td>COST</td>
<td>COST</td>
<td>COST</td>
</tr>
<tr>
<td>PAYDAILY</td>
<td>PAYEDIT</td>
<td>$241.56</td>
<td>$81.26</td>
<td>$138.21</td>
<td>$22.09</td>
</tr>
<tr>
<td>PAYDAILY</td>
<td>SORT</td>
<td>$534.87</td>
<td>$81.26</td>
<td>$453.63</td>
<td>$79.89</td>
</tr>
<tr>
<td>PAYDAILY</td>
<td>PAYUPDTE</td>
<td>$310.36</td>
<td>$156.23</td>
<td>$102.39</td>
<td>$51.74</td>
</tr>
<tr>
<td>PAYWEEKLY</td>
<td>PAYGROUP</td>
<td>$623.90</td>
<td>$98.45</td>
<td>$498.75</td>
<td>$56.70</td>
</tr>
<tr>
<td>PAYWEEKLY</td>
<td>SORT</td>
<td>$1,344.03</td>
<td>$246.80</td>
<td>$958.29</td>
<td>$138.94</td>
</tr>
<tr>
<td>PAYMONTH</td>
<td>PAYCHECK</td>
<td>$931.56</td>
<td>$105.90</td>
<td>$741.56</td>
<td>$84.10</td>
</tr>
</tbody>
</table>

The data center views program execution as the base unit for accounting, an approach that is not necessarily well understood by users and that offers little applicability to the business process.

The user of the payroll system, in this case the payroll department, would probably prefer to have its charges expressed in terms of the numbers of employees maintained in the system or the number of checks produced. For example, the payroll department could have been charged $36.45 per month for each salaried employee maintained in the payroll system. This transaction accounting approach provides a cost that is directly related to the business process and one that provides a stable means to control, evaluate, and plan for expenditures.

The batch payroll system is a straightforward illustration of the difference between existing resource charging approaches.
and transaction accounting.

ONLINE SYSTEM EXAMPLE

A more dramatic example is illustrated with the use of online systems such as TSO, IMS, or CICS. Consider the use of TSO for program development and the simple problem of updating and compiling a COBOL program under TSO.

Assume that the user of the TSO service, the application programmer, has made 25 source program update changes through the TSO EDIT sub-command facilities, then compiled the program interactively. Most data centers would consider the resources (CPU, I/O, etc.) used during the TSO terminal session as the base for charges. In other words, a user would be charged for all the processes completed during the terminal session, including the update and compilation work. However, this approach provides no breakdown of the types of work done by the programmer or the associated charges. The programmer thus has little cost information with which to evaluate and control the use of computing resources or the design of specific programs from a cost standpoint.

This example of TSO usage illustrates the advantages of transaction accounting. If a standard fee had been established for each TSO EDIT sub-command (for example, $0.03 per EDIT line change), resulting in a source program updating charge of $0.75 (25 EDIT line changes at $0.03 per change), each of the TSO EDIT sub-commands invoked could have been charged out. Additionally, because the processing of a COBOL compilation is a data dependent process, the COBOL source statement could be considered the transaction in this case. With this qualification, it would be possible for the data center to charge out interactive COBOL compilations on a per source statement basis (for example, $0.05 per COBOL source statement).
2.4 Transaction Accounting

2.4.2 Implementing Transaction Accounting

From all perspectives, transaction accounting can be an extremely advantageous method of accounting. However, the initial effort to implement a transaction accounting system, and the extensive continual administrative support, is expensive.

Organizations should consider the value and utility of transaction accounting. If the approach makes sense for your organization, you can plan for a gradual change to transaction accounting as new applications systems are phased into production. Effective transaction accounting is so different from existing charge-out system implementations that few organizations, if any, will be able to afford the luxury of retrofitting transaction accounting into computing processes that are currently in production.

When implementing transaction accounting, keep in mind that for a computing process to be a good candidate for transaction accounting, it must be composed of clearly defined transactions. While most computing processes do, at some point, possess this vital characteristic, a few processes do not lend themselves to this systematized method of charge-out. For example, in program development and testing, transactions can be defined for the assemblies, compilations, and links involved in the development work, but no well-understood input or output transaction has been defined for program testing. Therefore, program development is suitable for transaction accounting, but program testing is not.

Before you can use a transaction accounting system, you must have in place certain vital elements. These are discussed in the following sections:

1. Resource Measurement Systems
2. Cost Accounting Systems
3. Transaction Logging Facilities
4. Charge-out Systems
5. Cost Study and Administration
2.4.2.1 Resource Measurement Systems

Before a price or charge can be established in all business situations, the cost of manufacturing or production must be quantified by using resource measurement. For batch, TSO, IMS, and CICS, the computing system must be able to measure the computing resources consumed at the "event" level for the area (for example, batch program, TSO command, CICS transaction, IMS transaction). Note that IBM's Information Management System (IMS) and the Customer Inquiry Control System (CICS) are, from an accounting standpoint, analogous processes in that they are both message-driven systems.

The following minimum measurement capabilities must be operational:

- Batch Job Step Costing: Measure the usage of computing resources, including CPU, I/O, memory, device usage, DASD consumption, etc.
- TSO Command Costing: Measure the usage of computing resources, including CPU, I/O, memory, device usage, DASD consumption, terminal activity, etc.
- IMS/CICS Transaction Costing: Measure the usage of computing resources, including CPU, I/O, file I/O, memory, terminal activity, related supervisory functions, etc.

The resource measurement systems should provide the capability to:

- Count the executions of an event, such as the number of times that each specified TSO command is invoked.
- Measure in detail the resources consumed for each execution of an event, such as the CPU time, terminal I/Os, and amount of virtual storage used for the total number of invocations of the specified TSO command.

It may be prohibitive to always collect detail measures at the event level. The requirement for detail measurement does exist, however, and should be activated for a long enough measurement period to ascertain the resource usage of a given set of programs, commands, or transactions.
2.4.2.2 Cost Accounting Systems

Based on the usage statistics collected with the resource measurement systems, cost accounting computes processing costs to determine the cost of production. The costing of resources is a highly controversial area and will not be covered in this section. However, cost accounting provides data center management with quantified costs for processing batch programs, TSO commands, and IMS and CICS transactions. The important point for the reader to understand is that this step does not represent charge-out or billing, but only determines the cost of processing.

In addition to knowing the costs for the individual programs, commands, etc., the cost accounting system must have an "explosion" facility that can aggregate the processing costs of the jobs, programs, commands, transactions, etc., to determine the total cost of a particular application. The following explosion table illustrates this process:

<table>
<thead>
<tr>
<th>CATEGORY</th>
<th>IDENTIFIERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSO</td>
<td>USERID=PAYROLL COMMAND=PAYENTRY</td>
</tr>
<tr>
<td>BATCH</td>
<td>JOB=PAYDAILY PROGRAM=PAYEDIT</td>
</tr>
<tr>
<td>BATCH</td>
<td>JOB=PAYDAILY PROGRAM=SORT</td>
</tr>
<tr>
<td>BATCH</td>
<td>JOB=PAYDAILY PROGRAM=PAYUPDT</td>
</tr>
</tbody>
</table>

The cost accounting system calculates the processing costs for each of the three batch steps and the one TSO command for the accounting period, based on the usage quantified by the resource measurement system. By considering only the procedures specified in the explosion table, the cost accounting system then automatically accumulates all costs for the four specified functions to arrive at the total cost for processing the entire Payroll Check Generation Application.
2.4.2.3 Transaction Logging Facilities

Transaction accounting implies that the accounting process understands and has quantified the number of transactions that have been processed for the user. Unfortunately, this is not the situation in most data centers today. The computer charging systems have little or no knowledge of the number of transactions processed (checks generated, employee records maintained, mailing labels printed, etc.).

The following examples illustrate the ways a transaction can be the product (output) or service (input) as viewed by the user:

Batch:

System A produces a financial statement from general ledger expense entries. The expense entries, as input, represent the transactions (for example, $1.20 per general ledger expense entry serviced).

System B produces payroll checks and each output check is a transaction ($4.50 per payroll check produced).

TSO:

Command A requests the time of day. The transaction is the TIME command input by the user ($0.25 per TIME command serviced).

C L I S T B lists the records of a file and each line displayed is a transaction ($0.05 per line printed).

I M S or C I C S:

Transaction A requests that the salary of an employee be displayed. The employee salary request is both the transaction and the user input ($0.23 per employee salary request serviced).

System B maintains personnel data on all employees. Each employee maintained is a transaction ($1.76 per employee processed).
This accounting quantification cannot be measured in a traditional global manner (for example, by SMF), but must be calculated and logged by the application systems that are the processors of the transactions. In other words, the batch program that generates payroll checks within the Payroll Check Generation Application must inform the accounting system that 1,435 payroll checks were printed, and the TSO command that compiles a program must account for the fact that 325 source statements were processed.

If no one has built accounting facilities within the application processors to count and log transactions, a large measurement hole exists. Hence, we strongly recommend that you consider transaction accounting only as new applications are developed, and that you do not attempt to retrofit into existing systems.

For transaction accounting to become a reality, the data center accounting staff must provide the application systems developers with the methodology and the means for counting and logging transaction activity. Transaction logging requires facilities that:

- Allow the application program to invoke a routine that builds and logs an accounting record to the accounting collection file (for example, SMF).

- Allow the application program to invoke a routine that finds the task’s appropriate internal accounting record (for example, SMF step termination record type 4) and inserts the necessary accounting data into the existing record structure.

These facilities need a parameter-driven subroutine that is accessible by any programming language and the ability to provide at least two data elements: a code identifying the transaction being accounted (for example, 103 is for hourly payroll checks) and a count of the number of transactions processed (for example, 83,576 hourly payroll checks).

For transaction accounting to become possible, the need for transaction logging must be addressed.
2.4.2.4 Charge-out Systems

Once you have the components to collect usage statistics (resource measurement systems), compute processing costs (cost accounting systems), and identify and account for the transactions processed (transaction logging facilities), you are ready to consider the charge-out process.

This section assumes that standard fees are derived by adding the data center's cost ("cost" defined as direct cost plus an overhead burden factor) and a pricing markup.

Using the processing cost developed by the cost accounting system and the number of transactions, you can determine the direct cost of production.

In the case of the Payroll Check Generation Application, if the processing cost was calculated to be $389.45 for a three-month period, and in that period of time 236,188 checks were produced, it could then be derived that the direct cost was $1.65 per 1000 checks. When a 106% overhead burden factor is added, the cost per 1000 checks increases to $3.40 ($1.65 + 1.06($1.65)). With an 18% pricing markup, the final fee per 1000 checks is $4.00 ($3.40 + 0.18($3.40)).

The result is the objective of transaction accounting: A standard fee ($4.00) applied per transaction (payroll check) for a long enough period of time to ensure accounting stability (one year).

The above example illustrates what appears to be a rather simple situation. In many cases, transaction accounting may actually be this simple to apply. However, two processing patterns, as related to transaction accounting, should be considered in great detail:

- The length of the measurement cycle that is used to arrive at the "true" processing cost
- The nonlinear costs of application systems as they relate to the volume of transactions
LENGTH OF A MEASUREMENT CYCLE

One of the inherent dangers of a period measurement approach is that a periodic measure may not collect all the resource activity that is required to derive the average usage for the event. Consider an IMS transaction (Accounts Payable Scan) that is used to analyze all of the accounts payable entries in the Accrued Accounts Payable database. Because this database will vary in size with the time of the month (larger in the beginning and smaller at the end, directly after payables processing), low, normal, and high processing levels for an event must be measured. Indeed, the resource measurement system must measure the total cycle over a period of at least one full month to establish a true cost for the Accounts Payable Scan.

APPLICATION SYSTEMS HAVING NONLINEAR COSTS

Many application systems have processing attributes that result in either increasing or decreasing costs per transaction as the transaction volume increases. Costs may decrease as an application system takes advantage of economies of scale.

Unfortunately, increases in cost are more typical. Generally, as transaction volume increases, the processing costs increase more than proportionately; in other words, on a nonlinear basis. This situation can be caused by programming architecture, including internal table handling overflow, database expansion, string manipulation, list processing, index file overflows, etc.

The net result of either the increase or the decrease is that the transaction accounting standard fee per transaction must be slightly modified. The recommended approach is to provide step level standard charges based on transaction volume ranges. Consider the case illustrated below:

COBOL compilations with the source statement as the transaction:

- 00001 - 00400 standard charge per statement is $0.03
- 00401 - 01000 standard charge per statement is $0.05
- 01001 - 50000 standard charge per statement is $0.10
The step level standard charges have a great similarity to pricing techniques seen in many other facets of business. Consider the similarity to the photocopying business when examining a typical price schedule for a photocopier.

Each copy is one transaction.

001 - 010 Standard charge per copy is $0.10
011 - 100 Standard charge per copy is $0.05
101 - 300 Standard charge per copy is $0.025
301 - 400 Standard charge per copy is $0.02
401 - 999 Standard charge per copy is $0.01

Options include collating @ $0.25 per hundred copies and three-hole punch @ $0.30 per hundred copies.

Notice that in the copier price schedule there is a basic cost for handling the transaction (that is, copy) and additional standard charges for options.

The example below applies to COBOL compilations that use the source statement as the transaction. Source listings and cross-reference listings are options that increase the cost of a compilation. Thus, it is advantageous to have option pricing for compilations. The following table illustrates a potential set of charges:

<table>
<thead>
<tr>
<th>Source Statements</th>
<th>Basic Cost per Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 400</td>
<td>$0.03</td>
</tr>
<tr>
<td>401 - 1000</td>
<td>$0.05</td>
</tr>
<tr>
<td>1001 - 5000</td>
<td>$0.10</td>
</tr>
</tbody>
</table>

Additional charges per option:

<table>
<thead>
<tr>
<th>SOURCE</th>
<th>XREF</th>
<th>DMAP</th>
<th>PMAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0.005</td>
<td>$0.020</td>
<td>$0.005</td>
<td>$0.015</td>
</tr>
</tbody>
</table>

If a program was compiled that had 300 source statements and requested a SOURCE listing and cross-reference list (XREF), the total charge would be $16.50 ($9.00 base charge and $7.50 for the two options).

The benefit of this approach is that the charges are now directly associated with the selected options. You can estimate the cost before execution.
2.4.2.5 Cost Study and Administration

The many benefits and advantages of transaction accounting should be obvious by this time. Equally obvious, however, is the associated cost of implementation and administration. Implementing and administering a transaction accounting system is equivalent in scope to implementing and administering an industrial manufacturing cost accounting system. Consequently, the associated manpower and potential organizational changes required for transaction accounting must be considered. The following points address the organizational and political considerations of transaction accounting.

COST STUDY AND ADMINISTRATION GROUP

Before you decide to evolve or migrate to transaction accounting, you should realize the overall importance and impact of that decision on your organization. We recommend that you establish a separate functional entity, which reports to the Director of Data Processing, to administer the transaction accounting process. It is not necessary, or even desirable, to have the personnel who maintain the resource measurement and cost accounting systems under the auspices of this group; however, regardless of where the personnel for those systems report within the organization, the cost study and administration group should be responsible for establishing guidelines and objectives for those systems. The responsibilities of this group should include:

- Defining accounting policy
- Defining operational specifications for resource management systems, cost accounting systems, and transaction logging facilities
- Identifying transactions
- Determining the standard cost and charging rates
- Periodically auditing existing costs and charges
- Negotiating charging rates with the user community
If transaction accounting is to be effectively employed, it is essential to realize the importance of this undertaking and properly delegate the authority and responsibility to the cost study and administration group. This high level of control is necessary because, once transaction accounting is implemented, it becomes an integral part of the data center's operation and effectiveness.

STANDARD COST AND CHARGE PROCEDURE

As new applications are developed, an additional consideration is added to the application development responsibilities: accounting for the transactions (see Section 2.4.2.3, Transaction Logging Facilities). It is the responsibility of the cost study and administration group to develop the costs and charges of the transaction prior to production operation. This effort is no different from that used in manufacturing processes today where pilot systems are tested and costed prior to implementation.

The considerations that may be included in the standard cost and charge procedure are:

- Identifying and defining the transactions
- Defining and implementing transaction logging facilities
- Implementing the necessary resource measurement systems (for example, if this is the first IMS application, an IMS transaction-based resource measurement tool will have to be installed)
- Integrating the resource management system data into the cost accounting system, with the necessary administrative tasks addressed to establish the appropriate cost explosion tables (see Section 2.3.2.2, Cost Accounting Systems)
- Determining costs and charging rates, and negotiating with the responsible user groups
- Integrating the charge-out system

While the initial rate determination phase for establishing an accurate and representative standard charge is an extensive and arduous project, it is a crucial one.
PERIODIC COST AND CHARGE AUDIT

Standard charges will, over time, fall out of synchronization with their actual processing costs. This can easily result from any of the following reasons:

- Modifications in the processing techniques of the applications systems
- Changes in the "projected" processing volumes upon which the standard costs and charges have been based
- Changes in the software and hardware configuration used to process the application systems

In attempting to consider this problem, there are several solutions to evaluate, including:

- Automated variance analysis, which reports significant variation between the standard and actual costs on an exception basis to the cost study and administration group. In this way, the group has a predefined list of those transactions that may require a rate change.

- Periodic samples of charges and costs may be used to determine whether a significant variation exists between the standard and actual costs. If a sample does find such a variation, you should pursue further measures and evaluation.

- User-requested rate evaluation may be required when a user seriously challenges the standard charges set by the data center. In this case, the transaction in question would be completely reevaluated from a cost and charging standpoint to determine if there is a discrepancy.

Each of the above mechanisms provides the cost study and administration group a means to evaluate and calibrate the standard costs and charges that are in use.
2.4.3 Summary of Transaction Accounting

In conclusion, the following comments on transaction accounting are appropriate:

- Effective transaction accounting is based on a transaction definition of a data event that is well understood, identifiable, and pertinent to the user's process.

- Transaction accounting that considers only charges at the batch job or program, TSO command, or IMS/CICS transaction level provides necessary costing data, but does not provide a user-oriented standard charge.

- If an installation does not have effective resource measurement and cost accounting facilities, transaction accounting cannot be effectively implemented.

- The implementation of transaction accounting will most likely have far-reaching impacts on the data processing organization, its management, and the users of the data processing services, because dollars will be associated to understandable units such as payroll checks.

- The associated cost of implementation and administration must be well understood before undertaking such a large and expensive effort.

- The usability and benefit of transaction accounting demands that charge-out oriented installations at least consider the transaction accounting approach's feasibility.
2.5 Differential Charging

Differential charges are adjustments to the total charges for a piece of work. They are applied after all resource consumption has been calculated. The final amount is multiplied by a factor of greater than one or less than one, according to the type of differential charge required. Differential charges include two related areas: priority accounting and zone factors. Priority factors are applied before zone factors. The types of differential charging are discussed in the following sections:

1 - Priority Accounting
2 - Zone Accounting
3 - Product Surcharge Accounting
2.5.1 Priority Accounting

Priority accounting is the application of a surcharge for providing faster turnaround. One of the common non-data processing implementations of priority accounting is the Postal Service’s rate structure in which Express, Air, First, Second, and Third Class Mail, respectively, represent slower and less expensive services.

In a data processing environment, the common implementation of service level accounting is to define turnaround categories (in minutes or hours) with different charging factors. Figure 2-1 defines a simple priority or service level scheme:

<table>
<thead>
<tr>
<th>Turnaround Limit</th>
<th>Surcharge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less Than 10 Minutes</td>
<td>2X normal</td>
</tr>
<tr>
<td>Less Than 1 Hour</td>
<td>1.5X normal</td>
</tr>
<tr>
<td>Less Than 3 Hours</td>
<td>1.2X normal</td>
</tr>
<tr>
<td>Less Than 8 Hours</td>
<td>1.0X normal</td>
</tr>
<tr>
<td>Less Than 24 Hours</td>
<td>.75X normal</td>
</tr>
<tr>
<td>Less Than 48 Hours</td>
<td>.50X normal</td>
</tr>
</tbody>
</table>

Figure 2-1. Sample Priority Rating Factors

The reasoning behind priority accounting is that the more time a data center has available for scheduling work, the more likely it is to schedule it at a time when resources would normally be idle. Therefore, the longer the time allowed, the lower the rate, because the additional time allows the use of a product (the DP resource) that has a zero shelf life.

Priority accounting systems also tend to limit peak demand because users have an incentive to use off-peak hours. When the data center can spread the workload, it can reduce the need for additional installed capacity and ensure high utilization. The resultant reduced unit costs for work also tend to justify the priority accounting approach.
2.5.2 Zone Accounting

Shift accounting (zone accounting in CA MICS terms) is a surcharge or discount based on time of day. A zone surcharge is, in effect, a penalty charge for transacting business at the most desirable time. A zone discount is an incentive to transact business at a less desirable time.

The justification for zone accounting is that it tends to distribute the demand for resources over a wider timespan, thus reducing the peak demand and the corresponding capacity required to support it.

The potential problem with zone surcharges is that if the penalty is large enough, new peaks will form at the previous low demand periods. When this happens, it requires either a surcharge adjustment or a change in the relative surcharge applied to each zone. Either of these courses of action, however, may result in dissatisfied users.
2.5.3 Product Surcharge Accounting

Product surcharge accounting applies a charge to the users of a specific program product. The product is, of course, an expense to the data center that provides it to its users.

The rationale for product surcharge accounting is to recover the cost of a product from those who use it rather than including it in the overhead that is supported by all data center users. Where an expensive, special-purpose product is used by a small group of users, this philosophy has some merit. The alternative is to require the user group to pay the license fee for the product, but if it is possible that the product's use may become more widespread, the surcharge approach may be more acceptable.

A special case of product surcharge accounting occurs in the service bureau environment, where the service bureau pays a royalty to the product vendor for the use of a product. This royalty fee is passed on to the user in the form of a product surcharge.

There are several alternatives that may be used to recover the cost of program products from the applicable users.

Percentage Surcharge

The percentage surcharge is a multiplier of the standard rate for the use of the specified program product. For example, Job A ran program Product X. The resource charge for Job A was computed at $1.00. However, the Product Surcharge Table indicates that jobs executing Product X are subject to a 20% surcharge. Therefore, the submitter of Job A is billed $1.20.

Fixed Surcharge

A fixed surcharge is a flat dollar charge for the use of the program product. For example, Job A ran program Product X. The resource charge for Job A was computed at $1.00. However, the fixed surcharge table indicates that jobs executing Product X are subject to a $2.00 surcharge. Therefore, submitter of Job A is billed $3.00.
2.6 Additional Accounting Options

Subscription Surcharge

A subscription surcharge is a fixed monthly fee for the users of a specific program product. For example, User A has subscribed to use Product X and is therefore charged a monthly subscription fee of $75.00.

2.6 Additional Accounting Options

An accounting system can enhance its flexibility by offering different accounting options. These can include such items as charges for additional services, a selection of billing options, and different credit policies. The following sections discuss some accounting options:

1 - Support Services Accounting
2 - Billing Options
3 - IS Inventory and Assets Accounting
Most data center accounting concentrates on hardware usage that is directly related to a job or session. However, a large portion of data center costs are not directly related to hardware expenses. The major expense for data centers is personnel costs. Large service bureaus, whose only business is data processing, spend up to 70 percent of each revenue dollar on labor. For data processing operations that are internal to the company, personnel costs are typically one-fourth to one-third of total data center expenses. Obviously, a large portion of data processing costs are not equipment costs, but still need to be accounted for in costing data processing service.

The most common accounting approach for support costs in data processing is to assign each support person or cost to a specific hardware category. For example, tape operators are part of the tape cost center; computer operators are included in the CPU cost center, etc. The personnel or costs that are left over are usually called overhead and are allocated to all the direct cost centers based on some formula.

This approach is successful if sufficient care is taken to assign the support costs. It is usually possible to assign almost every cost to an appropriate cost center. Some examples of support service items for which an IS organization can charge its users include courier service, data entry, development work, training, printing, and the loan or exclusive use of hardware and software.

Obviously, the assignment of support costs affects the rate that would be charged to break even in a particular cost center. For example, if tape equipment costs $1,000 per week and 100 tape mounts were made in an average week, then mounts would appear to cost $10 per mount. But since $2,000 is also spent per week in labor for the tape operators, the true cost of the mount is closer to $30.

To obtain the most valid method of distributing costs to users, you should include the cost of support services in an IS accounting system.
2.6.2 Billing Options

An installation accounting system should support a number of options to increase user flexibility. These options are described below.

Fixed charges

In some installations, certain cost centers may be charged a fixed monthly fee for computer services regardless of their resource utilization.

Minimum charges

Some installations may wish to implement a minimum charge for a job or a session.

Discounts

In addition to service and zone discounts, an installation may give a global discount for a given cost center. For example, the Applications Development Department may be given a 30% discount on all of its work.

Surcharges

Surcharges are the opposite of discounts, giving an installation the option of adding a fixed percentage to the calculated charges either globally or for a given cost center. Surcharges may apply to the cost center's total charge or only to a specified class of charges (an invoice category).

Non-billable accounts

In some environments certain cost centers may be non-billable. That is, the cost of their usage is tracked but no invoices are generated for them, nor does their usage appear on the data center's profit and loss statement. A common example of a non-billable account is the data center's own technical support personnel.
2.6.3 IS Inventory and Assets Accounting

One of the major concerns of data processing executives is the financial management of the hardware and software used to satisfy the needs of their organization. The tracking of payments made to vendors, vendor invoice verification, depreciation, purchase option accruals, investment tax credits, and cost allocation are all part of the financial information that management must coordinate to achieve adequate organization profits and cost control.

An inventory and assets management system must have the ability to apply cost information effectively in a chargeback system. The inventory costs of particular components become the basis for direct charges to specific users. However, some equipment or software cannot be directly charged, but rather must have their costs recovered from resource, transaction, or other chargeable methods.

Cost recovery via resource, transaction, or other charge methods necessitates rate setting. One of the essential ingredients of rate setting is your organization's inventory and assets financial information.

To effectively manage inventory and asset financial information and relate it to data processing users, you must have a system that is capable of providing adequate information to a chargeback system. CA-I/S Inventory and Assets Manager (CA-IAM) allows you to maintain charging information that you can conveniently use with CA MICS Accounting and Chargeback.
Since data storage is one of the most expensive areas of IS organizations, chargeback to the user is crucial to the economics and planning of the organization. It is therefore necessary to have an accurate measurement of utilization so that users can be charged in a manner that encourages efficient space management.

The two basic methods of direct access space accounting are Volume Table of Contents (VTOC) snapshots and active monitoring. These methods differ in the accuracy of the data captured and the overhead necessary to capture it.

VTOC SNAPSHOTs

By far the most common method of collecting data for direct access space accounting is to read the VTOCs of the online packs at regular intervals, usually once a day, and to assume that the space utilization is relatively constant between "snapshots." There are two problems with this method:

- It does not account for temporary space utilization, which may be a significant load center for the installation; and,
- Because the snapshot is generally taken during the off hours to minimize performance impacts, it accounts for only the minimum space utilization level.

However, the VTOC snapshot approach has several things to recommend it:

- It is relatively inexpensive.
- It does not require coding of operating system modifications or exits.
- It does not incur the overhead of data capture for each Direct Access Device Space Monitor (DADSM) operation.
- It is easier to maintain than active monitoring.

This is the approach supported by the CA MICS Space Collector Option and CA MICS Space Analyzer Option.
ACTIVE MONITORING

An alternative to snapshot processing is the active monitoring approach. With this methodology, a software monitor determines space utilization by intercepting data set maintenance functions as they are performed. The keynotes of this approach are accuracy and the ability to account for temporary space utilization.

The problem with this approach is that the overhead of collecting the data may not be recovered by the additional accuracy that it allows, especially since the majority of the overhead occurs during periods of heavy system use. For this reason, it is not as popular as the VTOC snapshot method.

The following sections discuss the different types of DASD/DFHSM/SMS accounting:

1. Permanent DASD Space Accounting
2. Temporary DASD Space Accounting
3. DFHSM Migrated Data Sets
4. DFHSM Backup Data Sets
5. VSAM Accounting
6. Systems Managed Storage (SMS) Accounting
7. DASD Accounting Features
2.7.1 Permanent DASD Space Accounting

Accounting for permanent DASD data sets requires a system to identify the pertinent data set names and the space allocated to them. Then data set ownership is ascertained to identify the cost center to be charged. Data set names, standardized by a data set naming convention, are used to determine ownership. VTOC scans usually produce the necessary information.

VSAM data spaces, however, present a special charging problem. At sites that do not run IBM's Integrated Catalog Facility (ICF), VSAM naming conventions do not use the owner name in the data space name. Even when ICF is installed, the ICF Data in Index Component Name does not typically provide enough information for billing.

An additional complicating factor of identifying VSAM data set ownership is that a single VSAM data space may contain many clusters (data sets), which belong to separate users. Since VSAM catalogs contain unique entries showing the standard data set name for each VSAM cluster, the problem can be avoided by determining VSAM data set allocations by scanning the VSAM catalogs.

The information for non-VSAM data sets obtained by the VTOC scan and the information for VSAM data sets obtained from the VSAM catalog scans provide complete charging information for permanent DASD data sets. This combination of VTOC and catalog scans provides a standard approach for charging.

It is important to optimize the performance of both scans to minimize operational and overhead problems associated with gathering the data.
DASD charging uses space units such as volumes, cylinders, tracks, or bytes of data over a unit of time, such as days, to produce a charging unit of measure, such as track days (where one track day is equivalent to occupying one track for one day). The charging unit allows the DASD information to be extracted at varying intervals while keeping the same unit of reference. One representative unit of measure is the megabyte hour, which denotes the occupancy of one million bytes for one hour. The use of megabytes provides a consistent measure that does not vary with device type, unlike tracks, which are device dependent; the timespan of hour provides more precision than day or month. While the traditional track day qualified by device type is still a viable unit of measure, it does not allow for as much flexibility as megabyte hour. The charging units multiplied by a rate per unit produce a charge for each data set, which is billed to the appropriate cost center.

Small data sets present a dilemma when the charge for them is so small (less than one cent) that it rounds to zero. Three alternatives to address this situation involve calculating the charges on a monthly basis, using a minimum data set charge, or carrying greater precision in the charge field.

Another important DASD charging consideration is that some permanent data sets may be used by many cost centers. The charges for these shared data sets can be distributed among multiple cost centers by prorating the cost. The installation sets the guidelines for determining how the charges are to be distributed.

As IBM’s System Application Architecture evolves, it may be more reasonable to assign the cost of shared databases to a corporate account and view the data bases as a corporate asset. Another option may be to recover the cost with access charges to users.
2.7.2 Temporary DASD Space Accounting

Since temporary DASD data sets exist only for the duration of the task that allocates them, the VTOC scan technique cannot accurately capture their space allocations. Therefore, you should exclude your temporary data set volumes from the VTOC scan.

You can treat costs associated with the temporary data set volumes as overhead and recover them with the cost of the volumes for your permanent data sets.

Another approach is to actively monitor the space allocations when they occur. It can be implemented using the operating system's DADSM exits under Data Facility/Device Support (DF/DS) to monitor each temporary data set allocation, extent, or deletion to produce a record that contains the data set name, cost center(s), and space allocated or deleted. If DF/DS is not installed, you must make operating system modifications to provide "hooks" to capture the data.

The charging unit considerations for permanent data set charging also apply to temporary data sets. Note that cost center information for temporary data sets is usually derived from job or step level information (job name, accounting fields, and/or user ID) because temporary data set names do not contain sufficient information to identify the owner. In either case, the overhead of capturing and processing the data must be weighed against the benefits of charging for temporary data sets.
2.7.3 DFHSM Migrated Data Sets

Data in a DFHSM environment can be conceptualized as pyramidal in shape. This pyramid is meant to convey a hierarchy of storage ranging from PRIMARY or migration level 0 to migration levels 1 and 2. For simplicity, we abbreviate "migration level" to "ML" and thus, it is common practice to describe data as residing on primary, ML1, or ML2 storage. Primary is the normal location of data directly accessible by applications. When a data set has not been used for some specified period, HSM migrates it to ML1 storage volumes, where the data is usually stored in a compacted form and is not directly accessible. It can be recalled quickly to primary storage when needed. Finally, when installation thresholds have been reached, HSM will migrate data from ML1 DASD to ML2 storage. ML2 storage is usually a densely-packed tape cartridge.

```
     /
    /  \
   /    \
  /      \
 /_________
 Primary

     /
    /  \
   /    \
  /      \
 /_________
 ML1

     /
    /  \
   /    \
  /      \
 /_________
 ML2
```

In a typical DFHSM environment, the amount of data migrated through the above hierarchy could be presented as the above pyramid. For example, if 400 gigabytes of primary storage is available for active data, then perhaps a much larger quantity of data is inactive and has been migrated to less expensive levels of the hierarchy.
Information on migrated data sets (those at ML1 and ML2) is available from the CA MICS Space Analyzer Option HSM product via the CA MICS Space Collection Option. The CA MICS Space Collector Option (VCC) can optionally collect basic accounting and capacity planning information about data sets that have been migrated from standard VTOCs and moved to compressed locations in the hierarchy of storage managed by DFHSM. Locator information about migrated data sets is stored in the DFHSM Migration Control Data Set (MCDS). VCC serves as a “driver” for the collection of data from the MCDS.

Using the CA MICS Space Analyzer Option HSM and VCA products, you can establish different pricing for each level of the storage hierarchy. Using the pyramid again, you could conceive a rate structure that reflects the relative cost of each layer, as shown in the following example:

```
/\                      /
 / $ \                   / 0.10 \  
/ \ per \               / Mb/hr \ 
/____________\          
/ $0.05 \               / per \  
/ per megabyte hour \  
/ $0.002 \              / per \  
/ per megabyte hour \  
\______________________
```

See Section 2.7.7, DASD Accounting Features, for more information on this subject.
2.7.4 DFHSM Backup Data Sets

DFHSM can automatically create backup copies of data sets according to installation-defined rules. System Managed Storage (SMS) management class definitions can also be used to specify backup frequency and the number of backup versions that are to be kept. When DFHSM performs automatic backup of data sets, the Backup Control Data Set (BCDS) is updated with information about each backup copy or backup version created on backup volumes managed by DFHSM.

The pyramid forms a useful analogy for considering the potential amount of data for backup versions. Consider this pattern of usage for typical TSO users and the backup rules associated with a "STANDARD" management class:

- 25 percent of a given user's data sets are changed each day.
- An HSM rule states data sets are only to be backed up if they have been changed.
- A management class definition states that five backup versions are to be kept.

Thus, for an active data set that is changed every day, the BCDS will contain information for five backup versions of this data set in our hypothetical example above.

Expanding this scenario to an entire DASD complex, there might be the following kind of relationship in terms of the relative number of data sets:

```
/\  
 / \  
 /   
/ Primary \  
/       \  
/ Backup Versions \  
/_____________________________  
```
Information on backup versions of data sets is available from the CA MICS Space Analyzer Option HSM product via the CA MICS Space Collector Option (VCC). VCC can optionally collect basic accounting and capacity planning information about backup versions of data sets by serving as a "driver" for the collection of data from the BCDS.

Using the CA MICS Space Analyzer Option HSM product, you can establish different pricing for storage occupied for backup versions. Because the backup versions typically exist on high-density tape cartridges, the relative charge for storage occupancy (that is, amount of space times duration or megabyte days) is undoubtedly less for backup versions than for primary DASD storage occupancy. An example follows:

```
/     $1.25
/       per
/        megabyte day
/________________________________
/     $0.05
/       per
/        megabyte day
/________________________________
```

See Section 2.7.7, DASD Accounting Features, for more information on this subject.
2.7.5 VSAM Accounting

This section gives a brief overview of VSAM. It explains the difference between the VCA DAA and _VS files, why the _VS file used to be required for VSAM accounting, and why the DAA file is now preferred.

A brief overview of VSAM must limit itself to a very narrow part of this vast subject. The most important concepts related to accounting will be the boundaries of this discussion.

VSAM was first introduced by IBM in 1974 and has gone through several transformations since that time. For our purposes here, the most significant aspect of VSAM development occurred in 1981 with IBM's introduction of a completely new catalog structure called the Integrated Catalog Facility (ICF) catalog. The ICF catalog structure gathered acceptance at an increasingly rapid rate such that today most enterprises have totally converted to ICF structures.

A significant distinction must therefore be made pertaining to understanding accounting requirements for VSAM—namely, what the salient differences are between ICF catalog structures and the prior VSAM catalog structure which we will call "old-style VSAM" catalogs.

Old-style VSAM catalogs supported the concept of a "suballocated data space". This structure was represented by a single entry in the VTOC (that is, to MVS allocation routines, this appeared like one big VSAM data set). It was composed of free space and whatever VSAM clusters were defined within the data space (hence the term "suballocated"). The contents were only known by examining the catalog since the VTOC did not contain the "internal view" that was maintained by VSAM in the catalog.

From an accounting perspective, determining "ownership" from the VTOC by itself was impossible because the objects in the data space could belong to many different departments, divisions, or users. Again, only the catalog entries would allow a view into a suballocated data space.
ICF catalog structures require all VSAM data sets to be "unique" and hence, suballocated VSAM data spaces are not supported with an ICF catalog. The main impact of this implementation for our purposes is that there is now a one-to-one correspondence between an entry in the VTOC and an entry in the ICF catalog for a given VSAM data set. This means that the VTOC entry for a VSAM data set can be treated like any other non-VSAM data set for the determination of space allocated. The ICF catalog contains the space management information too, but it is redundant and it is not the definitive source of information about how much space is occupied on the volume. For example, the IDCAMS DELETE NOSCRATCH function can result in the VTOC becoming "out-of-sync" with the ICF catalog. Since such a scenario still leaves the space allocated on a volume, the VTOC is clearly the definitive source for accounting purposes. Likewise, if a data set is SCRATCHED but not uncataloged, the VTOC will reflect the fact that the space is no longer allocated. This allows unique VSAM data sets to be treated the same as non-VSAM for space allocation accounting. So while the ICF catalog will contain many things of interest to someone pursuing a performance problem or some other esoteric aspect of VSAM, it is the VTOC that becomes the most important data source for space allocation accounting information.

This leads to the conclusion that this relationship can be exploited to simplify your DASD accounting definitions to a single source—the VTOC. If your installation has converted to ICF catalogs, you should consider accounting for all data sets, including VSAM, from the VCADAA file that is built from the VTOC. The VCA_VS file is constructed from the ICF catalogs and any "old-style" VSAM catalogs if they exist. Treating the VCADAA file as the definitive source for information about space accounting is not only potentially more accurate but easier to understand since the distinction between VCADAA and VCA_VS for accounting purposes will no longer be necessary.

The design of the CA MICS Space Analyzer Option (VCA) is rooted in support for the suballocated spaces which were characteristic of the "old-style" VSAM catalogs. Support for these old structures is continued in VCC, but most installations have long since converted to ICF catalogs. You can determine if you have "old-style" VSAM catalogs connected to your systems by examining the output to the VCCSTATS ddname in a typical production run of VCC in your shop. Assuming you have only ICF catalogs (see the VCC User Guide for guidance in interpreting this listing), you should review
2.7 DASD/DFHSM/SMS Storage Accounting

Chapter 2: COST ACCOUNTING CONCEPTS FOR IS ORGANIZATIONS

your current methods of accounting for VSAM.

2.7.6 System Managed Storage (SMS) Accounting

One of the major enhancements incorporated in IBM's z/OS operating system is the logical implementation of System Managed Storage (SMS). By design, SMS isolates the end user from the physical implementation of the storage hierarchy. Because storage chargeback rates have typically been based on device type (DASD or tape) and quantity, often expressed in device-dependent terms (for example, 3380 track-days), SMS necessitates a major redesign of storage rate structures. Even more revolutionary, SMS changes the orientation of auxiliary storage management from the physical volume to explicit performance and availability criteria. This further changes the design of storage-cost recovery algorithms and the underlying methods by which the unit cost of storage is calculated.

For IS accounting, you must develop an effective method of directly charging for system-managed storage that does not depend on physical device-related characteristics. Furthermore, this method must be adaptable to auxiliary storage installations consisting of mixtures of diverse technologies (for example, conventional magnetic DASD and optical DASD) and implementations (for example, conventional DASD strings and DASD arrays). Your charging method must reflect the differences in the actual cost of these technologies while remaining independent of both the device and the technology. Otherwise, IS end users would have no reason to support the capital investments necessary to modernize.

Storage Hierarchies
-------------------

Modern computer systems implement storage in the form of a hierarchy in which each tier has a different ratio of performance to cost. At the top of the hierarchy is the high speed buffer; it offers the lowest access time but is so expensive that it cannot be very large given practical economic constraints. The next level of the hierarchy is main storage, followed by expanded storage, controller cache, DASD, and tape. Each of these levels trades off access speed for reduced cost.
The objective of this hierarchical structure is to provide a total performance level for the entire hierarchy nearly equal to that of the highest performance level at a cost close to that of the lowest performance level.

The lower tiers of the hierarchy, DASD and tape, are managed on a data set-by-data set basis, in an SMS environment, by DFSMS.

Data Set Placement
------------------

The primary consideration when placing a data set is usually the amount of space it requires. Because of the economic relationship between the layers of the storage hierarchy, the larger a data set is, the greater the incentive to place it at a lower hierarchical level to minimize storage costs.

The second consideration is frequency of access. If a data set is accessed frequently, the time required to mount a tape volume upon which it resides may become a significant portion of the total processing time associated with the data set. This is especially true if the data set is relatively small, because the processing time for a small amount of data will typically be correspondingly short.

Considered together, data set size and frequency of access determine the optimum media for a great many data sets. Large, infrequently used data sets belong on tape while small, frequently used data sets belong on DASD. In between these two boundaries lies a "grey area" in which additional considerations determine where the data set should be placed.

The third consideration for data set placement is the type of access. Tape, being a sequential medium, is poorly suited to data with random access patterns. Also, it provides little or no capability for inserting records into existing databases. Therefore, it is seldom a choice for IMS or DB2 data databases, regardless of how large they are or how infrequently they are accessed.
SMS Concepts and Constructs
--------------------

In a pre-SMS environment, data sets are allocated by storage users (application programs, TSO users, storage managers, etc.) on specific physical devices and possessing specific physical characteristics defined by the storage user (space required, method of allocation, logical record length, block size, retention, expiration action, etc.). In an SMS environment, data sets to be allocated are defined as having certain characteristics. These classes of characteristics are then mapped by the SMS components to logical pools of storage that is in turn mapped onto the physical storage hierarchy.

Data sets are assigned to storage groups based on the data's characteristics and storage requirements (performance, availability, etc.). These characteristics and requirements are divided into three classifications:

- **Data Class** - specifies many of the characteristics of the data such as record format, logical record length, etc.

- **Management Class** - specifies data management requirements such as length of retention, backup requirements, and disposition of the data set upon expiration

- **Storage Class** - specifies performance and availability criteria

By assigning a new data set to a specific management and storage class, the user defines the storage support requirements for the data. SMS can then map the request into the storage group that is best suited to meet those requirements. Subsequently, the physical placement of the data set can be further optimized as storage demands change or as different physical devices are added to the storage pool.
Life Cycle Management With SMS
----------------------------------

In manual storage management schemes, data is generally either active or archived. More often than not, archival of data (both onsite or offsite backup) is treated as an overhead of active data storage. Therefore, the unit cost of storage is simply computed as the fully-burdened cost of the physical storage (per unit of time) divided by the amount of available storage.

In an SMS environment, a data set may migrate from primary storage to secondary storage to archival storage during its life cycle. Therefore, the average amount of time a data set of a given management and storage class resides on each type of storage must be determined and the cost of the primary, secondary, and inactive storage must be computed individually and summarized together to arrive at the actual unit-cost of system managed storage.

Strategies for System Managed Storage Accounting
-----------------------------------------------

SMS technology requires a change from a device-dependent (for example, 3380 track-day) accounting algorithm to one based on space-time (for example, Megabyte-Hrs). Since SMS utilizes a class structure based on performance, availability, and data management requirements that were largely implicit in device selection, it seems logical to use these classifications as the basis for storage charging. They provide a basis for implementing a class-of-service algorithm that is equitable, easily managed by IS, and easily understood by the end user.

(Class-of-service charging schemes are widely utilized throughout service industries. For instance, the U.S. Postal Service offers a range of service classes, from overnight delivery to third class bulk rate. The major distinction between these classes is the speed with which the item arrives at its destination. These postage classes are analogous to SMS storage classes that determine how rapidly data may be accessed.)

In particular, SMS Storage Class defines service-level thresholds for performance and availability. These considerations largely determine the device type, cached versus non-cached, channel and control unit utilizations, etc., required by the data. These factors determine the
direct cost of the storage.

SMS Management Class defines the retention, backup and disposition requirements. These factors determine the indirect cost of the data storage.

Taken together, storage and management class imply both the performance cost and the support cost of a unit of data stored. This forms the basis for a class-of-service based charging algorithm as follows:

\[
\text{Data set charge} = \text{megabyte hours} \times \text{Storage Class rate} + \text{megabyte hours} \times \text{Management Class rate}
\]

Refer to the next section and Chapter 3 of the CA MICS Accounting and Chargeback User Guide for information on how to implement this algorithm at your installation.
2.7 DASD Accounting Features

This section summarizes CA MICS Accounting and Chargeback support for data sets that reside on active DASD volumes (primary storage), for data sets that have been migrated to migration levels 1 or 2 by DFHSM, and for backup versions.

This support allows you to implement the rate pyramid shown in Section 2.7.4.

CHARGING FOR ACTIVE DATA SETS ON PRIMARY STORAGE

CA MICS Accounting and Chargeback supports both SMS and traditional device dependent algorithms for DASD charging. This allows you to charge your SMS data sets by Storage Class and Management Class while you continue to charge your non-SMS data sets by device dependent measurements.

If you are currently using traditional algorithms to charge for SMS data sets, you can continue to do so, or you can implement Storage Class and Management Class algorithms for your SMS data sets without affecting the charges for non-SMS data sets.

CA MICS Accounting and Chargeback also supports VSAM accounting from either the VTOC or ICF catalog.

This support is provided by the use of three journal files, each with corresponding invoice categories and computation codes:

- The ACTJDA journal file is used for charging non-SMS data sets, both VSAM and non-VSAM, based on VTOC information.
- The ACTJMS journal file is used for charging SMS data sets by Storage Class and Management Class algorithms.
- The ACTJVS journal file can be used to charge VSAM data sets based on the ICF catalog, but this approach is not recommended.
CHARGING FOR MIGRATED DATA SETS

CA MICS Accounting and Chargeback supports charging for migrated data sets with the ACTJMG Migrated Data Journal file. This journal file is derived from the HSMMIG file and contains information on migrated data sets from the DFHSM Migration Control Data Set (MCDS).

The ACTJMG journal file allows you to charge different rates for data sets based on the migration level of the data, level 1 or 2.

You can also charge different rates based on the SMS Management Class value.

CHARGING FOR BACKUP VERSIONS

CA MICS Accounting and Chargeback supports charging for backup versions of data sets via the ACTJBC Backup Data Journal file. This journal file is derived from the HSBBAC file and contains information on backup versions of data sets from the DFHSM Backup Control Data Set (BCDS).

The SMS Management Class is often used to indicate the backup frequency of the data set and the number of backup versions that must be kept. Because data sets that are backed up more frequently, or require an extensive number of backup copies require more space and thus have a higher cost than do those with less stringent requirements, the ACTJBC journal file allows you to charge different rates for data sets based on the SMS Management Class value.

NOTE: An alternate approach to using the ACTJBC journal file to charge for backup versions of data sets would be to have proportional rates for data sets on primary storage based on their backup requirements (frequency and number of versions kept). Since the size of the backup versions will be proportional to the size of the data set on primary storage, if you charge for megabyte hours based on Management Class, and if you can identify the data set's backup requirements based on its Management Class value, this should provide a reasonable approximation of its backup costs.
2.8 Client/Server Accounting

There has been tremendous growth in the use of client/server systems, with most of the growth driven by the desire of businesses and government agencies to reduce the cost of computing by downsizing their applications to less expensive platforms. The platform chosen usually includes PC, Macintosh, or UNIX workstations. These are linked via local or wide-area networks to larger “servers” that perform database and communication services for the applications.

While the cost of the hardware, software, and support services for a client/server system may be less than the mainframe system that it replaced, it is still a significant part of businesses’ expenses. Fortunately, client/server systems can be accounted for, permitting managers to allocate and control their costs. This section describes the technology involved in client/server systems and suggests approaches to the problem of charging for their use.

Most large organizations have implemented accounting and cost control systems for their mainframe computer systems and the applications that use them. The same cannot be said of most local area networks (LANs). LAN costs are growing in most organizations and are often poorly accounted for. In the article “The Heavy Burden of LAN Costs,” Alan Radding, writing for Datamation, quotes a recent Forrester Research study that shows the average cost of administering the LAN in Fortune 1000 companies as $1270 per user per year. This is just the cost of administration and does not include hardware and software expenses. In many organizations, these expenses are poorly tracked and never reallocated back to the lines of business that they serve.

Because it is easier and cheaper to administer a few, large network servers than many smaller ones, networks share resources among many users. This makes it difficult to clearly link specific hardware and software to one department or line of business. Similar to their mainframe counterparts, client/server databases often serve more than just their primary owners. For example, allocating all of the costs of a customer database to a marketing or sales department would ignore its use by other departments such as customer service, shipping and billing.
When applications move from mainframes to desktop systems, accounting and control should go with them, but financial managers need new methods of chargeback to accompany the new platform. Mainframe systems have traditionally done job accounting based on resource usage: the number of CPU seconds used, pages printed, tapes mounted, and so forth. These same measurements are not always available for LAN systems and the “system” is often a collection of devices spread all around the organization, making data collection and summarization of usage statistics more difficult. There is also a growing movement among financial managers to reduce the costs associated with chargeback systems.

An accounting methodology is needed that is fair to all users, yet is easy to implement and does not impose too much overhead on the computer network or the staff that is assigned to administer it.

What is a Client/Server Application?

A general definition of a client/server application is one that splits the work between two separate components. Commonly, this means application logic that runs on a workstation supported by intense computing or database services performed by a larger computer. The client is the workstation program, the part of the application that the user sees and interacts with. The server stores and retrieves data and performs other services on behalf of one or more clients. The client and the server communicate over the network. So, each client/server application has three components: a client, a server, and the network that connects them.

The client/server application evolved as desktop systems became more powerful and less expensive. A modern desktop PC, Macintosh, or UNIX workstation is a powerful and flexible tool. Windowing software and graphical user interfaces (GUIs) have made applications easier to learn and use. The workstation is a natural platform for developing modern business applications, but affordable desktop systems have limited storage capacities and cannot share information with other systems unless they are linked together. Important business data must also be secured from unauthorized use and protected against accidental damage or deletion.
There Are Many Types of Servers
-------------------------------

A large PC, UNIX system, or mainframe computer can become the central device used for storing data that is shared among workstations. The choice depends upon how much storage is required, how many users will access the system and what software is required to provide the needed services. But data storage and retrieval are not the only functions provided by servers.

There are five common types of servers:

- File servers provide data storage and retrieval services allowing users to share files and keep files that are too large to save at their workstations.

- Database servers run database management software enabling shared access, security and recovery services for the data that they store.

- Print servers store output files that are waiting to be printed and allow many workstations to share a single printer or a set of printers.

- Compute servers run the intense computations required for engineering and graphics applications.

- Communication servers store and forward messages between users and between the client and server software. Special communication servers also translate messages between the formats used by the local network and the formats used by other remote networks or mainframe systems.

A single computer system could potentially provide all of these services, but in practice, services are usually separated to provide rapid response to user requests. By dividing its work, a network of small computer systems is able to provide service levels similar to those found on large mainframe systems. A single client/server application may use many servers to accomplish the tasks requested by its users. In a truly open system, a server may also become a client, requesting services from another server and then returning the final results to the workstation. All of this remains largely invisible to the user at a workstation, who sees only the results of this work being performed.
The Network
------------

The network is the third part of a client/server application and enables the clients and servers to communicate with each other. All communication between a client and server is in the form of messages. Clients send messages to a server to request data from a database or to request the server to execute program logic that resides on the server machine. Servers return answers containing the data requested or acknowledging that the action was carried out. The messages travel from one machine to another on coaxial cable, twisted-pair cable (telephone wire), radio broadcast, or light transmission. Special hardware and software provides the connections from the application to the network and between the various network devices.

National and international standards have been established to ensure that messages can be routed properly and interpreted correctly by devices and applications. The standards that govern the electronic signals used in the network define what is called the network architecture or topology. Networks are often known by the type of architecture that they use. Ethernet, ArcNet, and token ring are the three most common architectures for LANs. Mainframe systems have their own unique architectures. Organizations often standardize on a single architecture for their LANs, but connections to midrange or mainframe systems can add other architectures to the overall computing network.

The language used for messages is called a protocol. It determines how messages must be constructed by the software systems. The Transmission Control Protocol/Internet Protocol (TCP/IP) is the most common cross-platform protocol. But again, there are many others, such as the Novell IPX/SPX protocol used in NetWare LANs and NetBIOS used in Microsoft's LAN Manager and IBM's LAN Server. Communications software allows some networks to use more than one protocol, but where the language must change or the architecture is different, a special communications server called a gateway is used to translate the messages. Gateway connections are usually required between LANs and mainframe systems.

This combination of hardware and software is the network that links a server with its clients. It provides the communication essential for the application to function and may also contribute significantly to the cost of using the application.
The Applications Vary
---------------------

There are four types of client/server applications:

- Host-based. The majority of the application work takes place on the host or server machine and the workstation part of the application primarily presents the information to the user. A good example of this type is a graphical front-end to a mainframe application.

- Client-based. The application runs on the workstation and the server simply provides shared access to the data. Many database applications fit this category.

- Distributed. The application logic is distributed across the network on workstations and servers. Electronic mail is a good example.

- Ad hoc. The application runs on a workstation and may access shared resources such as databases and printers. The level of usage may be difficult to predict. Spreadsheet and personal database programs are good examples.

All four types of applications exist in most organizations. Each type uses the shared resources of a network differently. Our goal is to allocate the costs of providing these resources to the appropriate cost centers. The allocation method should be fair and consistent and should encourage appropriate use of the technology.

A Strategy for Client/Server Accounting
---------------------------------------

Accounting and chargeback for mainframe systems has traditionally relied on usage fees as the basis for application charges. Mainframe resource usage is measured in terms of CPU time used, thousands of I/Os performed, pages printed, and so forth. The associated charges are often billed directly to the user as line items on an invoice. Alternately, the user may be billed a fixed fee per month or a charge per business transaction, such as a fee for each check processed. In either case, the charge to the user is based on the computer resources consumed, either directly, on average per month or per transaction.
There are two reasons why usage-based charges are less desirable for LAN applications. First, the actual cost to provide the service is often determined by peak-usage requirements, not by total usage. Client/server applications move large amounts of data over the network. The size and cost of the server and the speed of the network are determined by the needs of the application. For example, graphic images are extremely large compared to text files, so applications that deal with graphic images must have high-speed networks. The second reason is that resource usage is not normally measured by user across a network. To collect usage data, it may be necessary to add hardware or software monitors throughout the network. Then, the data must be brought to a single point and accumulated. This adds a large overhead burden to the network and increases the cost of the service. It may also make the accounting system very expensive to operate.

For these reasons, organizations are adopting a chargeback system based on a set of subscription fees similar to the service charges used by the cable television industry. Each workstation attached to the network is charged a basic subscription fee that covers the cost of the network and use of common business-wide services, such as electronic mail and LAN backup service. Premium service fees are added for each user that may access other applications, services, or special devices that constitute a special level of service. This is often referred to as tiered pricing. Additional charges, based on usage, may still be added when applications access mainframe systems or high-end servers.

Identifying Service Tiers

The following guidelines may be used to determine what constitutes a service tier that will be charged a premium fee:

- The service or equipment is used by a minority of users and/or cost centers.
- The equipment, software, or overhead charges can be readily separated from other network costs.
- The total cost of the service warrants the added administration of tracking its users and invoicing for the service.
An Example of Charges

The following figures demonstrate how subscription fees for client/server applications would appear when combined with mainframe charges. Figure 1 shows a monthly invoice and Figure 2 shows a more detailed report of charges. In this example, the company has two distinct classes of users: the average corporate user with a PC connected to the LAN and engineering workers that have advanced workstations and special servers. The company has different base rates for these users: a "Basic LAN Service" fee for standard workstations and an "Engineering Workstation" fee for the high-end users. There are two premium services available: a news retrieval service used by some managers and a group of financial applications and databases used by finance, planning and selected managers.

Monthly Invoice
Manufacturing Division

To: Joseph Dokes
General Manager

<table>
<thead>
<tr>
<th>Category</th>
<th>Resource</th>
<th>Charges</th>
</tr>
</thead>
<tbody>
<tr>
<td>1XXX</td>
<td>Processor Hardware</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1100 Central Processor(s)</td>
<td>$1,206.80</td>
</tr>
<tr>
<td></td>
<td>1200 Channels</td>
<td>320.40</td>
</tr>
<tr>
<td></td>
<td>1500 Direct Access Storage Devices</td>
<td>739.98</td>
</tr>
<tr>
<td>2XXX</td>
<td>Teleprocessing Equipment</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2100 Terminals</td>
<td>490.60</td>
</tr>
<tr>
<td></td>
<td>2160 Control Units</td>
<td>181.00</td>
</tr>
<tr>
<td>8XXX</td>
<td>Facilities</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8600 LAN / Internet Services</td>
<td>4,803.00</td>
</tr>
<tr>
<td></td>
<td>Total Charges</td>
<td>$7,741.78</td>
</tr>
</tbody>
</table>

Figure 2. Sample Invoice
User Expense Summary for November 2005

To: Manufacturing Division  
ABC Electronics, Inc.

<table>
<thead>
<tr>
<th>Category</th>
<th>Resource / Description</th>
<th>Quantity</th>
<th>Units</th>
<th>Charges</th>
</tr>
</thead>
<tbody>
<tr>
<td>1100</td>
<td>Central Processor(s)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0022</td>
<td>Job CPU Time</td>
<td>6,191.50</td>
<td>TCB+SRB sec</td>
<td>$866.81</td>
</tr>
<tr>
<td>0041</td>
<td>Job Real Memory</td>
<td>1,241.42</td>
<td>K-core Hours</td>
<td>148.97</td>
</tr>
<tr>
<td>1500</td>
<td>Direct Access Storage Devices</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0092</td>
<td>Job DASD EXCPs</td>
<td>5,308.10</td>
<td>1000 EXCPs</td>
<td>265.41</td>
</tr>
<tr>
<td>8600</td>
<td>LAN / Internet Services</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8601</td>
<td>Basic LAN Service(s)</td>
<td>12.00</td>
<td>Stations</td>
<td>1,500.00</td>
</tr>
<tr>
<td>8602</td>
<td>Engineering Workstation(s)</td>
<td>18.00</td>
<td>Stations</td>
<td>3,150.00</td>
</tr>
<tr>
<td>8610</td>
<td>Financial Application Group</td>
<td>3.00</td>
<td>Users</td>
<td>105.00</td>
</tr>
<tr>
<td>8611</td>
<td>NEWS Service</td>
<td>1.00</td>
<td>Users</td>
<td>48.00</td>
</tr>
</tbody>
</table>

Figure 2-3. Sample User Expense Report

Summary

Client/server applications and the networks that support them are a growing expense for almost every business enterprise and there is an urgent need to apply the principles of management accounting to this area of expense. Because the technology used for these applications is very different from mainframe-centric computing systems, the approach to charging must be different also. Service-based charging is the recognized norm for client/server systems. It provides a simple and flexible means to transfer costs back to the cost centers that use the applications. The resulting service charges can be easily incorporated into existing chargeback systems.
The UNIX operating system is an increasing part of the enterprise computing solution. UNIX is finding its way into organizations for two reasons: maturity and openness. It is a mature operating system available for high-end client-server systems and it is open in nature, available on many platforms and from many vendors.

CA offers a solution to an organization’s need to integrate UNIX systems into their enterprise. The CA MICS Analyzer Option for MeasureWare (MW OS Agent Analyzer) and CA MICS Accounting and Chargeback enable state-of-the-art management of UNIX systems. Accounting data is integrated with mainframe, midrange and network charges, presenting a complete view of IS expenses to the user.

The CA MICS Analyzer Option for MeasureWare OS Agents processes data collected by HP MeasureWare. The resulting data, presented to CA MICS Accounting and Chargeback, allows users to charge for UNIX systems either by application category or at the detailed level of processes executed by each user.

The following section gives a brief overview of the UNIX system.

The UNIX System
----------------

The UNIX operating system was originally developed in the late 1960s through early 1970s by AT&T. Since then, AT&T, along with various other companies and major universities, has made significant additions and enhancements to the UNIX system. Currently, there are several versions of UNIX available in the marketplace: an updated version of the UNIX system from UNIX System Laboratories (USL, formerly a division of AT&T), operating systems derived from the USL UNIX system and new operating systems based on UNIX standards. It is the only computer operating system that has been implemented by nearly all major computer vendors and is available on systems that range in size from microcomputers to super mainframes.
Every UNIX system provides at least the following set of features:

- The UNIX kernel--In a strict sense, this is the UNIX system itself. The kernel is the operating system. It encapsulates the hardware and provides system services such as process management, memory management, I/O services and timer functions to application programs.

- A shell--The system interface between users and the operating system. Most UNIX systems provide more than one shell, including graphic shells such as the X Window System.

- A system library--Utility programs and system functions that may be invoked by users and applications.

The basic unit of work in a UNIX system is called a process. The UNIX system creates and destroys processes frequently. For example, when a user enters a command, the shell creates a process to execute the program, or executable file, associated with the command. If the command is made up of more than one executable file, a new process is created for each file that is executed. In an interactive UNIX application, each transaction is usually executed as a separate process, so a single user session may result in many UNIX processes.

The following sections describe the metric data available for charging and provide guidance in setting up PCS and MeasureWare parameters.
2.9.1 CA MICS Analyzer for MeasureWare

The MeasureWare system captures a broad range of performance and usage data for UNIX systems. It summarizes and stores the resulting metrics in compressed form in five log files: global log, application log, process log, device log and transaction log.

MeasureWare startup parameters may be entered that assign processes to an application name. The system administrator defines the application names and specifies one or more program names (executable files) and/or prefixes and, optionally, user IDs and priorities that make up the application. When a process is executed that matches the criteria, it is logged to the application. Log records are written periodically that summarize the activity for each application during the measured interval. Processes that do not fit in any defined application are summarized in the application named “Other.” “Other” is an automatic definition, you do not need to enter parameters for it.

The process log captures similar information, but it is not summarized by application. Startup parameters for the process log determine whether all processes or only selected ones will be logged.

CA MICS Analyzer for MeasureWare and CA MICS Accounting and Chargeback

The CA MICS Analyzer for MeasureWare prepares the MeasureWare data for CA MICS Accounting and Chargeback. With the CA MICS Analyzer for MeasureWare installed, two Accounting journal files, the UNIX/MeasureWare Application Journal File (ACTJMA) and the UNIX/MeasureWare Process Journal File (ACTJMP), are available for charging.

CA MICS Accounting and Chargeback allows you to charge for UNIX/MW resources from either journal file or a combination of both files. However, you should take care not to charge for the same work from both files.
The CA MICS Accounting and Chargeback Rate Table panels have validation checks to prevent duplicate charging in most cases, but they cannot ensure that applications selected for charging in the UNIX/MeasureWare Application Journal File do not contain processes selected for charging in the UNIX/MeasureWare Process Journal file. This can occur because the WORKUNIT field contains application name in one file and process name in the other file.

You can use qualification groups to easily charge for selected applications from the UNIX/MeasureWare Application Journal File and for any remaining usage from the UNIX/MeasureWare Process Journal File.

A qualification group defines a set of values for any one of the following accounting fields:

- **SYSID**—System Identifier.
- **Subsysid**—Subsystem. For UNIX/MW, this is the application number.
- **Subtype**—For UNIX/MW, this is the operating system identifier.
- **COSTCTR1**—Cost center 1 value.
- **WORKUNIT**—Unit of work. For the UNIX/MeasureWare Application Journal, this is the application name. For the UNIX/MeasureWare Process Journal, this is the process name.

Once you specify the individual values in a group in the qualification group table, you can use the group name in place of individual values to identify the work to be charged in a particular journal file.

For example, suppose you want to charge for application numbers 2 through 32 from the UNIX/MeasureWare Application Journal file, and for application number 1 (Other) from the UNIX/MeasureWare Process Journal file. You could define a Subsysid qualification group containing application numbers 2 through 32. Then you could use Rate Table Algorithm Qualification to specify this Subsysid group for the UNIX/MeasureWare Application Journal File and Subsysid value 1 for the UNIX/MeasureWare Process Journal File.
Metrics Collected for Accounting
--------------------------------------------------------

The following charging elements are available in the UNIX/MeasureWare accounting journal files:

UNIX/MeasureWare Application Journal
--------------------------------------------------------

<table>
<thead>
<tr>
<th>Comp code</th>
<th>Accounting code</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2601</td>
<td>APPCPUTM</td>
<td>CPU Seconds</td>
<td>Total CPU time</td>
</tr>
<tr>
<td>2602</td>
<td>APPSYSTM</td>
<td>CPU Seconds</td>
<td>System CPU Time</td>
</tr>
<tr>
<td>2603</td>
<td>APPUSRMTM</td>
<td>CPU Seconds</td>
<td>User CPU Time</td>
</tr>
<tr>
<td>2604</td>
<td>APPNICMTM</td>
<td>CPU Seconds</td>
<td>Nice CPU Time</td>
</tr>
<tr>
<td>2605</td>
<td>APPRTMTM</td>
<td>CPU Seconds</td>
<td>Real Time CPU Time</td>
</tr>
<tr>
<td>2606</td>
<td>APPNORTM</td>
<td>CPU Seconds</td>
<td>Normal CPU Time</td>
</tr>
<tr>
<td>2611</td>
<td>APPKDIO</td>
<td>1000 I/O</td>
<td>Physical Disk I/O</td>
</tr>
<tr>
<td>2612</td>
<td>APPKLRD</td>
<td>1000 I/O</td>
<td>Logical Disk Reads</td>
</tr>
<tr>
<td>2613</td>
<td>APPKLWT</td>
<td>1000 I/O</td>
<td>Logical Disk Writes</td>
</tr>
<tr>
<td>2614</td>
<td>APPKPRD</td>
<td>1000 I/O</td>
<td>Physical Disk Reads</td>
</tr>
<tr>
<td>2615</td>
<td>APPKPWT</td>
<td>1000 I/O</td>
<td>Physical Disk Writes</td>
</tr>
<tr>
<td>2616</td>
<td>APPPKBCT</td>
<td>Kilobytes</td>
<td>I/O Transfer Count</td>
</tr>
<tr>
<td>2617</td>
<td>APPKLI0</td>
<td>1000 I/O</td>
<td>Logical Disk I/O</td>
</tr>
</tbody>
</table>

UNIX/MeasureWare Process Journal
--------------------------------------------------------

<table>
<thead>
<tr>
<th>Comp code</th>
<th>Accounting code</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2651</td>
<td>KPSCPUTM</td>
<td>CPU Seconds</td>
<td>Total CPU time</td>
</tr>
<tr>
<td>2661</td>
<td>KPSKDI0</td>
<td>1000 I/O</td>
<td>Physical Disk I/O</td>
</tr>
<tr>
<td>2662</td>
<td>KPSKLDK</td>
<td>1000 I/O</td>
<td>Logical Disk I/O</td>
</tr>
<tr>
<td>2663</td>
<td>KPSPKBCT</td>
<td>Kilobytes</td>
<td>I/O Transfer Count</td>
</tr>
</tbody>
</table>

Figure 2-4. UNIX/MeasureWare Charging Elements

Some of the elements listed are not available on all UNIX systems. Refer to the “CA MICS Analyzer for MeasureWare” guide for the elements available on your UNIX system.
The recommended charging elements are CPU time and I/O transfer count. I/O transfer count is not available for all UNIX systems, however, in which case we recommend charging for logical I/Os. The UNIX system caches disk I/O, so applying a rate to logical I/Os provides a more consistent charge for applications and processes than physical I/Os. In the UNIX/MeasureWare Application Journal, the recommended computation codes are 2601 for total CPU time and 2616 for I/Os. In the UNIX/MeasureWare Process Journal, the recommended codes are 2651 and 2663. If I/O transfer counts are not available for your UNIX system, you can use computation codes 2617 and 2662, Logical Disk I/O, to charge for I/O usage.

Strategies for UNIX Accounting
----------------------------------------

MeasureWare summarizes metrics data into predefined application categories. This makes it possible to charge for resource usage without having to process large volumes of statistical data. To maximize this capability you must know, and define, what programs or program prefixes will be executed on a given UNIX system, and assign each of them to a particular application. Any undefined usage will fall into the catch-all application, Other.

In order to charge from the UNIX/MeasureWare Application Journal File, you must also be able to identify the cost center to be charged from the application name or number. This is because the application log does not identify the user of the application.

The UNIX/MeasureWare Application Journal contains fewer records than the UNIX/MeasureWare Process Journal, so charging from this file uses less overhead and is preferred. You can implement charging using only the UNIX/MeasureWare Application Journal if the following conditions are met:

1) All major processes are assigned to an application.

2) You can identify the cost center to be charged based on the application name or number.

3) The usage identified by the "Other" application can be treated as overhead, or assigned to a single cost center, or prorated on a percentage basis between a limited number of cost centers.
Another strategy, the one that organizations are most likely to choose, is to use the two journals in combination. Define major applications to MeasureWare and the accounting system. Charge for these applications from the UNIX/MeasureWare Application Journal. Charge for work that falls into the "Other" application category from the UNIX/MeasureWare Process Journal. The default UNIX/MeasureWare Rate Table definitions in CA MICS Accounting and Chargeback support this strategy.

If you have multiple UNIX systems, you do not have to implement the same strategy for all systems. You can specify the work to be charged in each journal file by system identifier (SYSID) as well as by application. This allows you to implement different strategies concurrently.

For example, suppose a company has three UNIX systems: one that supports a dynamic set of applications, one that supports a database with a fixed set of applications, and a third that supports both dynamic and fixed applications. For the first system you might charge by user from the UNIX/MeasureWare Process Journal, for the second by application from the UNIX/MeasureWare Application Journal, for the third by using both journal files, depending on the application number.

As a general guideline, select a method for each system that will provide the best accounting and reporting abilities using the fewest administrative and system resources. To do this, favor accounting by application; use charging by process when the applications and users change frequently.

Summary
-------

The MeasureWare software system collects performance and resource usage metrics on UNIX systems. The combination of MeasureWare from Hewlett-Packard with the CA MICS Analyzer for MeasureWare and CA MICS Accounting and Chargeback from CA provides an integrated financial management solution for users adding UNIX systems to the enterprise computing network. Users have the option to charge for resource usage by process, by application, or by using a combination of both methods.


2.10 DB2 Accounting

DB2 is IBM's relational database management system for the 
z/OS operating system. It was designed by IBM to enable 
users to develop and deploy applications in a consistent 
manner across a variety of platforms. With thousands of 
licensed users, it is one of the most popular relational 
database products in use today.

The DB2 application environment is technically very complex, 
but, with guidance, it is not difficult to implement cost 
accounting for DB2 applications. The greatest challenges to 
DB2 chargeback are to figure out the following:

- What resources to charge for, because users want 
  repeatability and predictability in their bills.

- What data source to select, because resource utilization 
  is reported by both DB2 and the application region.

- How to charge for DB2 overhead because much of the DB2 
  service usage cannot be attributed back to the user.

This section describes the DB2 application environment, 
discusses the technical issues involved in resource 
measurement, and defines a strategy for implementing 
accounting and chargeback. Using this information, you can 
easily implement the accounting strategy.

Background
----------

DB2 manages the relationships between the way an application 
program or user wants to access data and how the data is 
physically stored. It provides (almost) transparent access 
to data in distributed databases, guarantees that units of 
work completed by an application will be applied to the 
database, enforces security and data-integrity rules and 
makes intelligent decisions about how to access data 
efficiently for each request that it receives. In short, DB2 
handles much of the work needed to create database 
applications and lets the programmer concentrate on how to 
present or interpret the data and interact with the user to 
solve business problems.
DB2 users are either end-users, submitting queries from a terminal or workstation, or application programs running on mainframes or workstations. DB2 applications present a special set of problems for accounting and chargeback systems. This section describes the components of a DB2 application, provides the guidance needed to capture meaningful usage data, and recommends a chargeback strategy.

The DB2 System
---------------

DB2 has four major components, each of which executes as a separate region in the z/OS operating system. The first three are required while the fourth is needed only when remote database access is required.

- **System Services**--handles system-wide tasks such as system startup and shutdown, logging of data for restart and recovery, controlling connections to other z/OS subsystems such as CICS and TSO and collecting statistics for performance analysis and accounting.

- **Locking Services**--controls concurrent access to data, either permitting or preventing users from accessing the same set of data depending on the operations requested.

- **Database Services**--supports the definition, update and retrieval of data. It carries out the data requests from users or programs and manages the DB2 system tables.

- **Communication Services**--also called the Distributed Data Facility (DDF), is an optional component that implements distributed access to data. It communicates with DB2 databases running under z/OS, OS/2, or AIX and other non-DB2 database systems that use the IBM Distributed Relational Data Access (DRDA) protocol.

A DB2 system, composed of these regions, can "own" or control multiple databases and there can be multiple DB2 systems active in a single z/OS environment.
The Application Environment
---------------------------------

Users and programs communicate with DB2 using the SQL language. SQL statements are translated into program calls to DB2 that cause the various data functions to be performed. The SQL statements may be imbedded in programs that execute as CICS or IMS/DC transactions under TSO or in batch jobs. SQL statements can also be entered directly from TSO sessions and from PC workstations using one of a variety of front-end software programs.

If the application is running in the same z/OS system as the DB2 system that owns the database, then the call to DB2 is performed directly, using cross-memory services. If the data resides in a remote database, then the request is handled by DDF.

Measurement Data
----------------

A great deal of performance and statistical data is recorded for the application region and for the DB2 regions. The most useful statistics for charging for DB2 resource usage by an application or user are:

- Number of SQL statements executed by statement type
- CPU time used to execute the calls to DB2
- Number of getpage buffer requests used to process and retrieve the requested data

The number of SQL statements executed by statement type is available and may be used to apply a surcharge for DB2 usage. However, SQL statements vary greatly in the amount of DB2 resources that they use, so they are not a good choice for resource usage charges.

DB2 requests use TCB CPU time and the CPU time used to execute the user's tasks is recorded as TCB time. The TCB time used can be traced back to the user. Most I/O related activity and general DB2 services, such as locking and queue management, are in this category. This means that a significant part of the resources used in a DB2 application are recorded as overhead. They can be apportioned back to users but cannot be measured at an individual level.
The number of getpage buffer requests is a logical I/O measurement and a good indicator of the complexity of a DB2 request. It is the best element to use for recovering DB2 overhead costs.

TCB Time and Exceptions

TCB, or user-task, time for DB2 functions is recorded by the calling region. This is usually the application region: batch, TSO, IMS/DC, or CICS. It is therefore, normally accounted for by standard usage charges. However, there are two exceptions to this rule: CICS transactions and remote database calls.

o CICS Considerations

After DB2 Version 6, and CICS CTS Version 2.2 and later, CICS transactions use the Open Transaction Environment (OTE) to connect with DB2. This means that the DB2 TCB time is reported in the CICS transaction records and billing from the CICS application region will account for DB2 TCB time.

Prior to DB2 Version 6 and CICS CTS Version 2.2, CICS DB2 calls were executed under a separate z/OS subtask and a separate TCB from the rest of the transaction. CICS statistical routines did not track this time for a transaction, resulting in the TCB time used by a CICS transaction to perform DB2 functions not being recorded in the CICS transaction data.

o Remote Database Calls

Data requests that are passed to a remote database by DDF are executed in a DB2 region on the system where the database resides. This TCB time is available only in the resource statistics produced by the DB2 system that executed the request.
The TCB time recorded in the DB2 statistical file for local database requests by batch jobs, TSO sessions, IMS/DC, and CICS (after DB2 Version 6 and CICS CTS 2.2) transactions duplicates the time recorded by the calling region.

To accurately charge for TCB time used by a DB2 application, you must apply charges to the following elements:

- TCB time recorded in the batch job, IMS/DC, TSO, and CICS statistical files.
- TCB time recorded in the DB2 user activity file if the caller was a CICS transaction (but only if you are running DB2 Version 5 or earlier, or CICS CTS 2.1 or earlier).
- TCB time recorded in the DB2 user activity file for remote database access. For CA MICS Accounting and Chargeback Option, these records have a value of "1" in the DB2THDTY field.

The DB2 Accounting Trace
----------------------------

DB2 statistics are collected by the DB2 Instrumentation Facility, a part of DB2 System Services, and may be written to the SMF log or to a Generalized Trace Facility (GTF) data set. The facility allows users to select several types of trace records and the level of detail. The traces are Statistics, Accounting, Performance and Global. The level of detail is controlled by the Class of data requested. For more information, see the "DB2 Performance Monitor Usage Guide".

For accounting purposes, we are interested in the Accounting trace, recorded in SMF type 101 records. These provide the detailed information needed for accounting and chargeback. They contain the TCB time used, counts of DB2 calls by type and logical I/O counts. When the CA MICS Analyzer Option for DB2 and CA MICS Accounting and Chargeback Option are used, these statistics are available to the CA MICS Accounting and Chargeback Option system as charging elements derived from the DB2 User Activity (DB2DSU) file.
The DB2 Accounting Trace may be activated for various classes. The DB2 installation default is Accounting Class 1. However, Class 2 should be used instead to provide accurate TCB times for DB2 calls.

NOTE: There has been much debate about using Class 1 or Class 2 trace data for measuring CPU time. Most recommendations from IBM have stated that Class 2 time is a more accurate measurement of the "in DB2" time used for a DB2 call. The additional overhead required for the Class 2 trace has been significantly reduced since DB2 Version 3. For a detailed discussion of Accounting Class 1 and Class 2, see IBM's "CICS DB2 Guide."

Identifying DB2 Overhead
------------------------

At the system level, SMF type 30 records for the DB2 regions identify the resources utilized by DB2 that cannot be traced to an individual user. This includes CPU time, memory used, and other measures associated with standard job accounting. These statistics are available to CA MICS Accounting and Chargeback Option as charging elements derived from the BATJOB file. The charges associated with these elements for the DB2 regions are usually summarized to a no-bill cost center as a record of overhead costs.

Most organizations recover the overhead costs in charges to their users. The overhead may be recovered either by spreading the costs across all billed elements, thus sharing the burden among all system users, or by applying a charge to DB2 users only.

If the overhead is spread across all of the billed elements, then the overhead costs are included in the rates charged for CPU time, EXCPs, and so forth and no additional charges are added for DB2 usage. In this case, DB2 users and applications will be charged for the TCB time used for DB2 requests (at the standard rate) and other charges that apply to the charging elements recorded for their local region. No surcharge or add-on rate is applied for DB2 services.
If the overhead is charged only to users of DB2 services, then an additional charge is added for DB2 usage. In this case, the user is charged the standard rates for resources used in their region, including the TCB time used for DB2 calls, plus a DB2 usage rate that is applied to elements collected in the DB2 statistical file. Getpage buffer requests is the preferred element for the add-on charge.

Strategies for DB2 Accounting
-------------------------------

We recommend that you charge for DB2 resource usage based on Class 2 TCB time from the DB2 statistical records. If you are running CICS CTS Version 2.2 (or higher) and DB2 Version 6 (or higher) then you should NOT charge for CICS CPU time, and instead ONLY charge for remote data request CPU time.

There are other alternatives that may be more appropriate in your environment. For example, if a DB2 system is dedicated to a single application or business area, then the simplest charging method is to ignore the DB2 usage statistics and charge for DB2 resource utilization as a whole from the batch job statistical file.

Another alternative is to charge each user a fixed price per month for using an application.

Or you may be able to do transaction rather than resource-based charging. Using transaction charging, you charge users based on the number of business transactions processed. An example is, charging based on the number of customer bills produced by an accounts receivable system. This type of charging benefits the end users because it enables them to relate their information-technology costs to their actual lines of business and thus better plan expenses. For more information about implementing transaction charging (also called transaction accounting), see section 2.4.

The rates for transaction and application charging are nearly always based on the measured and rated cost of system resources used by the transaction or application. Once you establish a cost factor or rate for the resources used, it may be relatively easy to translate the charges into values that are more meaningful for the end users of the services. Thus developing rates for resource usage is beneficial for most charging strategies.
Tying DB2 Statistics Back to the User
-------------------------------------

In order to charge for DB2 usage from DB2 records, you must identify the user to be charged. There are several ways to approach this.

The statistical records written by the DB2 Accounting trace seldom correspond one-to-one with the statistical records from batch jobs, TSO sessions, CICS or IMS/DC transactions. For batch jobs and TSO sessions, there will be one or more DB2 records. For CICS and IMS/DC transactions, the DB2 statistics may represent part of a transaction or a combination of several transactions. Therefore, we do not recommend trying to match the DB2 records back to the records generated by the caller in order to identify the DB2 user.

Instead, we recommend using authorization ID (AUTHID) and the DB2 plan name to identify the DB2 user and assign the cost center to be charged. For TSO sessions, these fields normally identify a user and application area; for batch jobs, CICS and IMS/DC usage, they should at least identify the application used.

NOTE: This depends on an organization's use of the AUTHID and DB2 plan structure. It is assumed that the AUTHID used for TSO sessions is unique to the user or work group and that DB2 plans do not include data access modules from multiple application areas.

In addition to identifying the user, you also need to know the calling environment (batch, TSO, CICS, etc.) and whether the record represents a remote database access. The DB2 statistics should be summarized to minimize the amount of data before calculating charges, but the fields needed to identify the user, calling environment, and thread type must be used as summarization keys.
Using the CA MICS Analyzer Option for DB2, you can control the level of summarization based on the population of the DB2DSU account fields because charging occurs at the DAYS timespan. The contents of these fields are assigned during CA MICS Analyzer Option for DB2 installation; the recommended assignment statements are provided in Section 3.2.1, Implementing DB2 Accounting, in the CA MICS Accounting and Chargeback Option User Guide.

Summarizing charges in this way is most efficient for DB2 applications as well as the accounting and chargeback system. Performance analysis can still be accomplished at a fine level of granularity using statistical data in the CA MICS Analyzer Option for DB2's DETAIL timespan files.

Summary
-------

Although the DB2 application environment is complex, it is not difficult to implement cost accounting for DB2 applications. The TCB time used for DB2 calls is the only directly measurable element available for standard usage charges. For remote database (DDF) environments, TCB time reported in the DB2 Accounting trace must be added to the time reported for the application region. For CICS transactions, TCB time reported in the DB2 Accounting trace is added only for older releases of DB2 and CICS (DB2 pre-Version 6 and CICS CTS pre-version 2.2). DB2 overhead may be spread across all system users, both DB2 and non-DB2 users or may be applied as an add-on cost to DB2 users only.

The methods that have been described are easy to implement using the CA MICS Analyzer Option for DB2 and CA MICS Accounting and Chargeback Option and add little additional overhead to the chargeback system.
2.11 Network Accounting

Using network data sources for accounting allows direct recovery of the cost of the data communications network rather than its being absorbed as an overhead.

There are several types of algorithms that are available for network accounting. The use of any algorithm must reflect the IS organization’s structure, which may or may not be reflected in the design of the network. This section discusses three chargeback scenarios for networks with identical topologies but different organizational structures.

THE EXAMPLE NETWORK

In the following examples, the network consists of two sites as shown in the figure below.

```
T -------- T
\        / ----/-- T
 |      |  |    |
 |      |  | HO5T ----|-- ---- T
 |      |  |    |
 |      |  |    |
 |      |  |    |
 |      |  |    |
 |      |  |    |
 |      |  |    |
 T        T
```

SITE A        SITE B

Site A has the host processing resources and a communications processor (37xx equivalent). Site B has only a communications processor. Both sites have terminals attached to the network.
Data Processing as a Centralized Service for the Organization

In the data processing as a centralized service scenario, data processing is seen as tightly controlled. The central communications staff makes the decisions concerning the type of service, level of service, network topology, line speeds, terminals utilized, and so on.

Communications costs are allocated to end users based solely on the volume of data transmitted. The total communications costs are divided by the total traffic load to produce a cost-per-byte factor, which then applies to all traffic. Typically, the cost-per-byte factor is adjusted on an annual basis.

Since the user makes no decisions concerning the line speed, routing of transmissions, or service level of the applications, these factors are not taken into account in setting rates. It is assumed that the design of the network maximizes the cost/performance ratio as defined by the organization planners.

Data Transmission as a Centralized Resource

In the data transmission as a centralized resource scenario, data processing is viewed as a resource to be shared equally by all entities within the organization. The organization's communications network provides the connection to this shared resource. Typically, the user is responsible for the costs of the boundary circuits, terminal control units, modems, and local support personnel.

Costs of the network are allocated based on traffic volume and distance from a virtual host processor resource that is located equidistant from all users (despite the fact that the real data center is located at Site A). Charges may also be based on a standard network link speed and may be adjusted for varying link speeds.
2.11 Network Accounting

GUEST USERS

In the guest users scenario, Site B is a paying guest of Site A. Site B, which may be an acquired subsidiary of Site A, gains some advantage from using Site A's processing resources.

Site B bears the total cost of the data transmission to Site A, regardless of the distance. Site A does not subsidize Site B's communications costs, as in the example of the centralized resource, even though Site A may choose, install, and support all of the communication equipment.

Costs are allocated by totaling the costs of communication and dividing them by Site B's traffic volume to arrive at a cost-per-byte factor, as in the example of the centralized service.

SUMMARY

In each of the above examples, only the actual communications costs are recovered (modems, terminals, control units, links, support personnel, test equipment, etc.). Host processor costs are still recovered based on an application's consumption of host resources.

The differences in algorithms result from the interpretation of which portion of the resources belongs to the organization as a whole and which portion belongs to the end user.
Chapter 3: COST ACCOUNTING AND RATE DETERMINATION

One of the most important factors for the success of a profit-making manufacturing or service organization is its ability to determine accurately the costs of the products or services that it markets. Such costing information is the basis of product pricing as well as the foundation for evaluating the performance of the organization.

The philosophy behind the running of a profit-oriented organization should be applicable to a businesslike IS organization. It, too, should use valid cost accounting principles in determining and distributing the data processing costs to the users of its services. The service (or "product") of the IS organization is the output of information as produced from raw data supplied by the user, within a time-frame acceptable to the user. Using generally accepted cost accounting disciplines will produce:

- Data processing costs that can be properly assigned to each IS cost center (that is, user), enabling users to improve their management of data processing expenditures.

- Summarized costs permitting upper management to evaluate and plan the activities of the IS organization more effectively.

The major problem in establishing effective cost accounting procedures is applying cost accounting to a heterogeneous set of services. In an IS organization there are many different services in addition to computer processing services that are chargeable to the user. In fact, cost categories such as technical personnel, special forms, tape reel storage, reference manuals, technical education, and software acquisition may comprise the major portion of the overall data processing costs. Thus, each resource category should be individually costed and charged out accordingly. To include the costs of these other resources in the overhead to computer processing costs results in an inequitable cost distribution system and, more importantly, fails to provide true "responsibility accounting."
Responsibility accounting itemizes the data processing charges to the responsible user for each data processing service. Users know exactly where the money is spent, and can become more conscious of their computer and data processing resource utilization. This procedure provides both user and IS management with an accurate picture of how their data processing costs have been distributed, thereby permitting the type of priority decision making that characterizes effective management.

Conventional cost accounting systems are normally defined as being either process cost or job order cost systems. The process cost system is normally associated with the automated assembly line operation and is typified by the production of a single product as part of a continuous process, such as the manufacture of an automobile engine. This type of cost accounting system does not meet the requirements for the IS organization.

According to Patrick S. Kemp on pages 107 and 108 of Accounting for the Manager, published by Dow Jones-Irwin, Inc., 1970, the job order cost system is used for a job shop environment that produces many special products of a heterogeneous nature. This cost system has a structure that is applicable to the costing of data processing resource usage.
On pages 106 and 107 of the same book, Mr. Kemp identifies three ingredients used in job order cost systems to arrive at product costs: direct materials, direct labor, and manufacturing overhead.

Materials and labor are called direct if the costs of the items can be directly associated with the cost of the final product. All other costs, normally termed indirect costs, are considered to be overhead and cannot be associated directly with the cost of a product. This structure is only partially valid for data processing costing.

Like manufacturing, data processing costing relies on direct materials, direct labor, and overhead. However, because of the method of costing the computer equipment used, data processing uses a fourth cost category. In manufacturing, the actual equipment cost cannot be easily associated with the cost of a product and is normally considered as an overhead item. In data processing costing, the equipment usage can be directly measured and is associated with the end service. Therefore, the fourth cost category, direct hardware costs, is used in the data processing costing structure. In summary, the data processing costing system should have four categories of cost: direct hardware, direct labor, direct materials, and data processing overhead.

This section contains the following topics:

3.1 Load Center Definition (see page 118)
3.2 Load Center Cost Determination (see page 119)
3.3 Direct and Indirect Classification (see page 120)
3.4 Direct Cost Determination (see page 121)
3.5 Indirect Cost Application Methods (see page 122)
3.6 Pricing Strategy Selection (see page 125)
3.7 Determining Billable Rates (see page 129)
3.8 Periodic Variance Analysis (see page 130)
3.1 Load Center Definition

A load center is a single type of costing unit. In a data processing costing system, a load center might be represented by resource items such as a central processor, a pool of tape drives, a pool of EAM equipment, or a data entry department.

One objective of the data processing cost accounting process is to determine accurately the total cost of operating each of the load centers, considering their direct costs and associated overheads.
3.2 Load Center Cost Determination

You should consider the following steps in structuring the load center costs for the IS organization:

1. Identify the individual load centers that comprise the total IS organization.

2. Determine whether each load center is direct or indirect (see Section 3.3, Direct and Indirect Classification).

3. Define the rate that will be used to distribute the enterprise's overhead against the IS organization (that is, the burden rate).

4. Calculate the cost of each load center by following this procedure:
   - Determine all direct costs of hardware, materials, and labor.
   - Determine all indirect costs of hardware, materials, and labor, and define how each of these items is to be distributed against the direct costs.
   - Determine what other indirect costs might be distributed against this load center, short of including the overhead represented by the load centers defined as indirect.

5. Apply the corporate burden rate against each of the load centers based on your organization's accounting practices.

6. Evaluate how each indirect load center should be distributed as overhead against each of the direct load centers.
3.3 Direct and Indirect Classification

A load center is classified as either direct (operational) or indirect (overhead). A direct load center is one that can be charged directly to a specific work cost, such as the storage of tape reels in the tape library storage load center. An indirect load center, such as the services provided by the hardware/software selection group, cannot be charged directly to any specific work cost and must be included as a data processing overhead.

There is no fixed rule for classifying activities as either direct or indirect for all IS organizations. Generally, if the load center's cost can be directly tied to the work cost, then it is a direct center; if it cannot, it is an indirect center and is treated as an overhead cost.
3.4 Direct Cost Determination

From an accounting standpoint, it is always preferable to recover costs based on direct measurements rather than overhead charges, because the overhead charges are based on estimates. Direct measurements eliminate guesswork and have the additional benefit of continuing to maintain their accuracy when usage varies as a result of system or environmental changes. To encourage cost management, the accounting system uses direct costing as its primary costing method.

Direct data processing costs are categorized as direct labor, direct hardware, and direct materials. The load center to which these are assigned depends on the structure of the IS organization.

Whenever personnel work exclusively for one load center, the labor dollars are assigned to that load center. For example, computer operators are assigned to the CPU load center, tape librarians to the tape load center, and a DASD space manager to the DASD load center.

The accounting system can also assign a management position to a direct load center if the position is assigned solely to the load center. A first-line manager in the tape library whose work is solely concerned with the tape load center can be costed to the tape load center. Managers whose work supports several load centers should be assigned to an indirect load center. If you have information from a time reporting system that permits time breakdowns among load centers, you can use it to apportion labor costs for those personnel who support several load centers.

The load center distribution of direct hardware and software costs are best derived from a product such as CA-I/S Inventory and Assets Manager. Whether manual or automated, as a product enters the organization it should be tracked as to its use among the different load centers. Most products will have one load center to which they belong. Others will have to be distributed as an indirect cost.

Materials such as tapes and other supplies can also be costed directly to the load center using them, or, if their use is not directly attributable to one load center, to an indirect load center.
3.5 Indirect Cost Application Methods

You must evaluate the many different types of indirect costs individually to determine how they should be distributed and to determine the items against which they should be distributed. The IS organization applies a “burden rate” against the users that represents the organization’s indirect costs. If indirect costs are not distributed on an equitable basis, management does not have an accurate picture of the cost of operating each load center, nor is the IS organization able to distribute its costs to its users equitably. The following example illustrates the evaluation of indirect items:

The IS organization of company XYZ has three departments: A, B, and C, each of which is considered a load center. Load centers A and B are direct centers, while load center C is an indirect center. Figure 3-1 illustrates a sample set of costs for load centers A, B, and C.

<table>
<thead>
<tr>
<th>DEPARTMENTAL COST BREAKDOWN</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIRECT</td>
</tr>
<tr>
<td>Labor</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>Dept.</td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>C</td>
</tr>
<tr>
<td>All</td>
</tr>
</tbody>
</table>

Figure 3-1. Departmental Cost Breakdown
While there are many different methods of distributing corporate indirect costs, this example uses a burden rate that is a percentage applied against the labor costs of each division. The burden rate is 110%. (It would have been feasible to use either the number of personnel or total costs as distribution bases.) Figure 3-2 adds the burden rate to the total costs listed in Figure 3-1.

**TOTAL BURDEN RATE APPLICATION SCHEDULE**

<table>
<thead>
<tr>
<th>Dept.</th>
<th>Labor</th>
<th>Burden Cost (Labor + 1.1 * Labor)</th>
<th>Material/ Machinery</th>
<th>Total Cost + Burden</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>12,000</td>
<td>25,200</td>
<td>65,000</td>
<td>90,200</td>
</tr>
<tr>
<td>B</td>
<td>33,000</td>
<td>69,300</td>
<td>4,000</td>
<td>73,300</td>
</tr>
<tr>
<td>C</td>
<td>7,000</td>
<td>14,700</td>
<td>0</td>
<td>14,700</td>
</tr>
<tr>
<td>All</td>
<td></td>
<td></td>
<td></td>
<td>178,200</td>
</tr>
</tbody>
</table>

Figure 3-2. Burden Rate Application Schedule

To arrive at the final costs of the direct load centers, A and B, the cost of load center C must be applied to the two other load centers. Remember load center C is an indirect cost and must be applied against direct load center costs. It is also important to realize that even though load center C is an indirect load center, the burden rate was still applied to it to determine the accurate cost of load center C.
Load center C is a support activity maintaining materials and machinery. Therefore, it would be logical to distribute the cost of load center C across centers A and B based upon their material and machinery costs. The total cost of materials and machinery for the two centers is $69,000, with A responsible for 94.2% and B responsible for 5.8% of the total. In distributing the cost of load center C, the amount of $14,700 is distributed proportionately, with $13,847 going to A and $853 going to B. The final direct costs of the A and B load centers are shown in Figure 3-3.

<table>
<thead>
<tr>
<th>Dept.</th>
<th>Burdened Cost</th>
<th>Distributed Indirect Cost</th>
<th>Total Load Center Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$90,200</td>
<td>$13,847</td>
<td>$104,047</td>
</tr>
<tr>
<td>B</td>
<td>73,300</td>
<td>853</td>
<td>74,153</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$178,200</td>
</tr>
</tbody>
</table>

Figure 3-3. Total Load Center Costs

The object of this example is to illustrate two points: first, that there are many different types of indirect costs; and second, that there can be many different ways of distributing these costs. In defining indirect costs and distribution techniques, we strongly suggest that the cost accounting department be brought in as consultant to the data processing costing staff while the accounting system is being designed. If this participation with the cost accounting group is successful, the costing policies of the data processing system should adhere to those of the overall corporation. Such compliance with corporate procedures will go a long way in winning acceptance by both user and top management of the data processing costing procedures.
3.6 Pricing Strategy Selection

Accurately determining the operating cost of the load centers is only part of the accounting process. The critical consideration is in determining just how these costs are going to be charged back to the users.

In making this decision, you should consider the break-even, capacity, profit, and market-based pricing approaches. Your organizational environment will determine which among these is the most applicable philosophy. Then you can determine billable rates.

BREAK-EVEN CHARGEBACK TECHNIQUE

The break-even chargeback technique charges users of data processing services an amount equal to the IS organization's cost for rendering those services. Often referred to as a variable rate system, this method distributes the IS costs among the users for the accounting period. Each accounting period results in a different level of use, which causes the charges to the users to vary. The break-even chargeback may occur monthly, quarterly, or even yearly. The longer the fiscal time period, the less variable the charges for the users.

The break-even approach is the most commonly used method, but it can have some serious disadvantages from the corporate standpoint. There is no direct incentive for the data processing organization to minimize costs. They know that whatever they spend will be charged back to the users, allowing them to maintain their break-even position. This approach can result in large, centralized, computer complexes replete with an abundance of computing power, sophisticated software, and technical staff that uses significant amounts of money.

In using the break-even approach, the organization must estimate the level of utilization that each load center will realize. This level of utilization could then be divided into the load center cost figure to arrive at the unit rate that would be used to charge for services rendered by this load center. To illustrate this situation, assume that for a fiscal year the cost of the tape drive pool is $120,000 and the projected usage for the year is 6,000 tape hours. The unit rate to be used to recover this load center’s costs will be $20.00 per tape hour ($120,000 / 6,000).
CAPACITY CHARGEBACK TECHNIQUE

The capacity chargeback technique assumes that a unit rate for charging is not based on estimated utilization, but is based instead on the estimated capacity of the services. The only time break-even occurs is when the load center is operating at full capacity. In other words, this approach considers a budgetary loss to be a normal situation. The important control concept is that the level of loss really reflects either a waste or an underutilization of resources. By identifying the wasted resource, the system provides management with a new and valuable piece of information.

Additionally, the users of data processing services are not penalized for inefficient data center operation or excessive hardware and software capabilities. This type of reporting system provides an incentive for IS management to keep its costs down, maximize the utilization of current hardware and software capabilities, and discourage unnecessary additions in the form of hardware, software, and personnel.

The example of the tape drive pool also illustrates the capacity chargeback method. Remember that with capacity chargeback, the estimated capacity of the load center should be used, instead of the estimated utilization level. Also note that the estimated capacity used by this system is quite different from a load center's theoretical capacity. To support this, consider the following:

There are 10 tape drives on a system. In a 24-hour period there is a theoretical capacity of 240 tape hours. In reality, no resource is ever used to its theoretical capacity. The potential or actual capacity of a load center is what must be determined.

To arrive at the actual capacity, all down or unavailable time would first have to be deducted from the theoretical capacity. In addition, based upon the workload of the installation, it must be determined how much of the load center could be used if it were utilized at a capacity level. In determining this unused load center time, the time from schedule conflicts, resource conflicts, and overhead requirements must also be deducted from the theoretical time to arrive at capacity time.

$\text{Capacity time} = (\text{Theoretical time} - (\text{Unavailable time} + \text{Conflict time}))$
It might be determined that the actual capacity of the tape drive pool is, for example, 18,000 tape hours or three times the level of utilization projected. This actual capacity level of 18,000 would result in a unit rate of $6.67 per tape hour ($120,000/18,000). If the actual utilization was 6,000 tape hours, $79,980 would be designated as direct loss due to the level of under-utilization. This difference between budgeted and recovered costs provides management with an effective cost control and evaluation tool.

PROFIT CHARGEBACK TECHNIQUE

The profit chargeback technique allows the IS organization to plan for a profit at the end of the fiscal period. Service bureaus use this technique. An internal IS organization can also add a profit margin to the costs and assess its users. It can use this profit for expansion of user services or merely as a measurement tool. A profit technique also increases the user community's awareness of the strategic importance and business nature of the IS organization.

MARKET-BASED PRICING

A variation of the profit technique is the market-based pricing method. This approach views the IS organization as any other profit-making data processing organization. Thus, IS management sets its rates competitive with other vendors that provide computing products and services. It also supplements basic data processing with services such as technical support, application development, technology migration, and technical training.

When the enterprise's management allows users to purchase the same services from vendors outside the organization, the IS organization is forced to set its prices low enough to be attractive to customers who now have an alternative. While prices must be low enough to be competitive, the IS organization must also provide the other key factor for attracting users: a high standard of service. Users will purchase the in-house "product" only if the service that accompanies it is comparable to or better than what an outside vendor will provide.
Implementing this pricing policy requires a comprehensive management commitment. The new rate structure should be developed and reviewed according to management's objectives and financial resources. Rate setting should focus on lowering prices enough to attract a high volume of business. Among the features that an organization can offer are volume discounts, flat rates for dedicated use of resources such as DASD, and special rates for technology conversion.

The fundamental actions that an organization needs to take for a successful market-based pricing policy are the following:

- Obtaining detailed cost accounting
- Continually assessing the cost effectiveness of available products and services
- Setting realistic prices compared to other vendors
- Staying at the forefront of technological change
- Providing a consistently high level of user services
3.7 Determining Billable Rates

You can use the following steps to determine billable rates for each data processing service. Determining the operation costs for each load center should precede any rate determination.

1. Identify the unit element within each direct load center for which the user will be charged (for example, for the tape library load center, the unit element to be charged would be the storage of a tape reel per day, say $0.05 per reel/day).

2. Select the type of chargeback approach that should be used. The three primary methods are profit, break-even and capacity chargeback.

3. Determine the fiscal time period that will be used for the current set of rates being developed. We strongly suggest that you use at least one quarter, or if possible, six months or one year.

4. Project either the estimated utilization level or the capacity level, depending upon the chargeback method you choose. Approximate the utilization level by reviewing the historical usage and then relating that pattern to the projected workload growth or change. For capacity charging, derive the figure that represents the capacity level for this load center service. The capacity level must be represented in terms of the unit elements defined in step 1. (For example, for the tape library there may be 300,000 storage units with each unit representing the storage of one reel of tape for one day.)

5. For the period selected, determine the load center cost to be recovered.

6. Calculate the unit rates for each of the data processing services being offered.

In summary, two points must be stressed: first, understand what the actual costs are; and second, establish an equitable means for distributing those costs.
3.8 Periodic Variance Analysis

Since rates are typically based on a prediction of future costs and utilization levels, a periodic variance analysis is an important method for validating the rate structure. This variance analysis should analyze actual costs using the cost determination model you selected and compare the actual cost to the forecast. Use the same process to analyze the resource utilization levels. The two types of analyses should then prepare you for the next rate-setting exercise. New accounting systems should make sure to use a periodic variance analysis to help set and validate the rate schedule.