NITC Customer Handbook

Last Updated October 2003

United States Department of Agriculture
Office of the Chief Information Officer
National Information Technology Center
8930 Ward Parkway
Kansas City, MO 64114-3363

Table of Contents

Chapter 1. NITC Services

Introduction

The National Information Technology Center (NITC) is part of the United States Department of Agriculture's Office of the Chief Information Officer. Our Centers of Excellence provide Enterprise Solutions for products, services, and resources to complement information technology missions of the United States Department of Agriculture (USDA) and its agencies, the Federal Aviation Administration (FAA), General Services Administration (GSA), and many other government clients. The NITC headquarters is located in Kansas City, Missouri with other Centers of Excellence in Fort Collins, Colorado, Washington, D.C. and Beltsville, Maryland.

Our Centers of Excellence specialize in providing Total Enterprise Solutions utilizing government and industry standards. Our computing facilities have "leading edge technology" products, develop solutions using "best of breed" industry services. NITC offers resources for business continuity capability, consistency, and reliability.

The programs and applications that run in the NITC environment are national in scope and impact. Our mission is connected to the success of our customers by providing reliable and cost effective Information Technology (IT) solutions to achieve effective mission performance and program delivery for NITC clients. In addition, NITC has extensive experience in data center consolidations, workload transfers, migrations, web farms, and conversions.

The NITC team is committed to excellence in providing world-class leadership through innovative, reliable IT services and solutions in support of our valued customers. Our employees maintain a high level of professionalism and strive to provide superior customer service.

In addition, NITC has extensive experience in data center consolidations, workload transfers, migrations, and conversions.

Mainframe Services

The NITC operates six IBM Complementary Metal Oxide Semiconductor (CMOS) mainframe computers at its facility in Kansas City, Missouri. Four of the six mainframes are in a shared environment and two are customer owned. All enterprise servers hosted at the NITC operate under Open Systems Architecture (OSA) operating systems. This type of operating system offers a separate environment where developers can benefit from the portability of the UNIX world while taking advantage of the reliability, security, scalability, and availability of enterprise architecture.

Mainframe Processing Schedule

Normal schedule for USDA shared environments:
Sunday 2400 (CT) through Saturday 2400 (CT)
With the Exception of these times:
Saturday 2400 (CT) through Sunday 0600 (CT) for Disaster recover backups for all DASD volumes
Sunday 1600 (CT) through Monday 0500 (CT) Reserved for IPL and system maintenance.

Our technical staff support over 500 Commercial Off-the-Shelf (COTS) products to fill the information technology needs of our customers, including:

- Utilities Software
- Data Storage Products
- Teleprocessing Software
- Language Compilers and Related Products
- Database Management System Software
- Web Services Software and Tools (including WebSphere)
- Data Warehouse Software and Tools
- Transaction Processing Software
- Automated Schedulers and Production Control Systems
- Telecommunications Subsystems and Protocols
- Decision Support Software
- Office Automation Software
- Text Processing Software
- Statistical Packages
- Executive Information System Software
- Human Resource Management Software
- Computer-Based Training (CBT)

NITC centralized services include:

- COTS software management
- System software development and certification
- Capacity and performance management
- Procurement
- Security Administration
- Storage Administration
- Contingency Planning and Disaster Recovery
- Backup and Recovery
- Technical Support

Mid-Range Services

The NITC hosts a growing mid-range environment. The NITC partners with customers to provide services leveraging the strength of mainframe technology in the mid-range environment. We offer a centralized computing facility for mid-range platforms, software and support.

NITC will house and manage client/server computer systems for customers on an individual basis through an agreement package that includes: Task Order, Reimbursable Agreement, and Memorandum of Understanding (MOU).

Platforms supported include IBM RS/6000, Hewlett Packard 9000, SUN SPARC systems, and Pentium-based servers. Operating systems supported include HP-UX, IBM-AIX, Sun-Solaris, and MS Windows.

Standard service levels have been established by the NITC for our mid-range services. Contact 1-888-USE-NITC (1-888-873-6482) for more information.

Applications Development

Applications development services are provided by the Agency Application Services Division (AASD), a NITC Center of Excellence located in Fort Collins, Colorado. AASD offers contract services for application development and support in all phases of the system life cycle, as specified in reimbursable agreements.

AASD develops mainframe and distributed IT applications for its clients; provides advice relating to development, implementation, maintenance, and operation of application systems; and offers a large range of other information technology services, such as, requirements analysis, specification development and project management.

Our staff possesses extensive expertise in database and applications life cycle support. We work in partnership with customer agencies to assure a comprehensive and integrated approach to applications development and maintenance by incorporating all areas of systems, database, applications, and documentation disciplines.

AASD delivers a variety of quality products and services, including:

- Database and applications design and development
  - Oracle, Sybase, Informix, System 2000, RDB, Microsoft Access, dBASE IV, and R:BASE
  - Database and Website development and integration (E.G., Cold Fusion, NetDynamics)
  - COBOL, Fortran, C, C++, Pascal, Perl, Java, Java Script, Motif
  - Oracle Designer and Developer 2000
- Integration and testing across multiple platforms
  - IBM/MVS, AIX, X Windows, MS Windows, and Windows NT

- IBM Mainframe, UNIX Workstations (IBM RS/6000, SUN, HP)
- Honeywell Bull, Data General, DEC VAX

- Database and applications maintenance and support
  - Database administration
  - Applications maintenance
  - Help Desk
- Website development and management
  - Webmaster
  - Site and page design and development
  - Requirements analysis
  - Development of site/page standards
  - Website backup
  - Usage reports
- Database and Web Technical Documentation

Applications services costs are recovered through service rates for billable hours of labor as cited in reimbursable agreements with client agencies.

Contact NITC

To find out more about NITC services call **1-888-USE-NITC** (1-888-873-6482). Our mailing address is:

USDA-OCIO-NITC
P.O. Box 419205
Kansas City, MO 64141-6205

For applications development services, contact the Fort Collins Office at **(970) 295-5210**. The mailing address is:

USDA-OCIO-NITC
200 Centre Avenue
Fort Collins, CO 80526

The NITC Customer Support Center (CSC) is manned by the System Network Control Center (SNCC) and provides Level 1 problem resolution available on a 7x24 basis. All IT Specialists at NITC are considered Level 2 support and may be contacted by the SNCC when needed outside of normal duty hours.

Help Numbers

(816) 926-6681	*NITC Customer Support Center*
(816) 926-6660	*System Network Control Center*

Policies

Contacting Vendors - Customers wishing to contact NITC hardware or software

vendors may obtain points of contact information from Center technical personnel or the Customer Support Center, if appropriate. NITC technical personnel must initiate the call and forward problem descriptions and other pertinent data to vendors for resolution unless the customer has a contractual agreement with the specific vendor.

Vendor Manuals - The NITC does not acquire or distribute vendor-supplied manuals. Each agency is responsible for ordering vendor publications they require.

References

Additional information about the NITC may be obtained by accessing http://www.ocio.usda.gov/nitc/index.html or www.agbiz.usda.gov/.

Chapter 2. Security Policies

Introduction

The National Information Technology Center (NITC) provides operational security through hardware, systems software, physical and personnel security measures, and the implementation of backup and recovery disaster operations. Protection is provided through logon procedures, auditing of security, and backup and recovery procedures.

NITC Security Features

The infrastructure maintained by the NITC includes Uninterruptible Power Supply (UPS) and diesel generators, which provide an independent source of power.

Checkpoint Firewall and VPN software products are used to provide network perimeter protection and Virtual Private Networks (encryption) for client/server and S390 environments.

Network Intrusion Detection Servers (IDS) provide real time data on events as they occur. Vulnerability scans are performed on all servers monthly to provide the data needed to protect systems from the most current attacks. Scanner and ISD systems are updated consistently. Emerging security tools and strategies are continuously researched and implemented.

All staff and contractors are required to have a limited background investigation and depending on the individual's access and job duties, clearances up to top secret are required.

Access to mainframe systems at the NITC is controlled by CA-ACF2 (Computer Associates - Access Control Facility 2) security software. Individual users access the services of the Center using a Logon ID (LID) with a changeable password.

What a user is allowed to do is controlled by access rules that are written within the controlling software by the Center (general permissions) and the Agency Security Officer (specific permissions). Generally, all installed software products are available to all users unless an Agency Security Officer dictates otherwise.

Certain abilities, functions, commands, programs and databases are tightly controlled by the Center and access is granted only in response to a written request from the head of the office through the Agency Security Officer.

Mainframe Security

Logons and Catalogs -

It is the general policy of the Center that no single user be issued more than one logon. When a new logon is desired, the Agency Security Officer should be contacted and the following information provided to the Center:

LOGONID (LID)
NAME (person assigned)
PHONE NUMBER (including area code)
ACCESS DESIRED (TSO, Non-TSO, CICS, etc…)
ACCOUNTING CODE (Agency Security Officers control this)
DISTRIBUTION CODE (Agency Security Officers control this)
TELEVIEW PROFILE (Agency Security Officers control this)
CICSID (for CICS only)
DEFAULT DESTINATION (Remote ID) (where appropriate)

Access Rules -

It is the responsibility of the Agency Security Officer within the security software to build, compile, and store (new rule or modification) a rule for each user logon. Without this rule, access to the system by a user will not be allowed. The NITC Security Team cannot build this required rule, as it has no knowledge of the permissions the Agency Security Officer wishes to allow.

The Security Team also will not answer questions from individual users about the structure of their, or any other, rule. The Center's Security Team does not take action against any user rule. This is the responsibility of the individual Agency Security Officer.

Initial Logon -

Each new logon is assigned an initial password that the Agency Security Officer will make known to the individual user. This password will allow an initial access but will require modification before additional processing is permitted. The user will note and respond to the following messages at the initial sign on:

1. ACF01017 PASSWORD FOR LOGON HAS EXPIRED (the new logon id)
2. ACF82008 ACF2, ENTER NEW PASSWORD
3. ACF82020 ACF2, REENTER NEW PASSWORD FOR VERIFICATION
4. ACF82000 ACF2, LOGON IN PROGRESS
5. ACF01129 PASSWORD SUCCESSFULLY ALTERED

The user will then receive three asterisks (***) from the system and should depress the enter key for normal processing.

If a user changes the password but the changed password is not matched when verified, the system will allow the user seven (7) additional tries to verify the new password before terminating the session. If terminated, the user should merely sign on again to change and verify the new password.

If the user is unable to sign on, it usually is due to one of following reasons:

- logon has not been properly created.
- high-level qualifier (catalog) has not been properly created.
- proper access rule has not been written/stored.

The Agency Security Officer should be contacted. That officer can write or correct the rule, or the officer will contact the Center to verify the correctness of the logons and catalog. Only the user's Agency Security Officer should be contacted. The Center's Security Team will not respond to individual users.

Reset Logons -

When logons are suspended due to unmatched passwords or when a user cannot remember their password, Agency Security Officers can reset logons by resetting the password. This password is good only for an initial entry. When the password is reset, the entry procedure and messages are identical to those for new logons as outlined above.

Unmatched Password -

If a password is entered improperly, the messages and actions are as follows:

1. ACF01012 PASSWORD NOT MATCHED
2. ACF82004 ACF2, ENTER PASSWORD

The system will allow the user three (3) chances to enter the password correctly.

After three failures, the system will send the following message:

ACF82904 ACF2, SESSION TERMINATED

Users are allowed another three (3) chances to enter the password correctly (a total of six attempts in all). After six (6) attempts, the logon is suspended. The Agency Security Officer must reset the logon.

Expiring Password Message -

The security system will allow a password to be used only for a period of thirty-five (35) days. Starting on the 31st day, the following message will appear at each logon:

ACF01134 YOUR PASSWORD WILL EXPIRE ON MM/DD/YY – HH:MM

Users may wait until the password has expired and then enter a new password as outlined above, or they may change the password as outlined in the following section.

Changing Passwords On-line -

User may change a password before it expires as part of their normal logon procedure.

1. ACF82882 ACF2, ENTER PASSWORD –
   ** User responds by entering both the old and new password in the format *oldpassword/newpassword* **
2. ACF82020 ACF2, REENTER NEW PASSWORD FOR VERIFICATION –
   ** User reenters new password in the format *newpassword* **
3. ACF01129 PASSWORD SUCCESSFULLY ALTERED

Re-using Passwords -

The security system prohibits the reuse of the current password and three (3) previous passwords. Any attempt to reuse a password will result in one of the following messages:

> NEW PASSWORD CANNOT BE THE SAME AS CURRENT PASSWORD
>
> NEW PASSWORD MATCHES A PREVIOUS PASSWORD

The current password must be kept for a minimum of four (4) days, ninety-six (96) hours before it can be changed again. An attempt to change the password any earlier will result in the following message:

> NEW PASSWORD NOT SET – CURRENT PASSWORD MUST BE KEPT FOR 4 DAYS

Security Contacts

The Security Team of the NITC responds only to the Agency Security Officer (or Deputies) of the agencies doing business with the Center. An Agency Administrator has identified Officers and their deputies in writing.

While calls from individual users are welcome, the Center's Security Team usually is unable to respond to them, as the security policies of the several agencies are not known to the Center. Individual users should direct all action and/or informational requests to their Agency Security Officer(s) who will direct them to the Center, if necessary.

Chapter 3. Computer Operations

Introduction

National Information Technology Center (NITC) Computer Operations strives to provide a reliable computing environment, with services to our customer community, including functions related to connectivity, data processing, job scheduling, systems monitoring and operational performance.

The System Network Control Center (SNCC) provides first level customer help desk assistance for problems ranging from access and connectivity to queries on jobs and/or data processing performance, utilization and automation. . All jobs and processing submitted to the Center are processed as expeditiously as possible.

Hardware and related equipment are monitored, maintained and secured, in a stable environment created by such environmental control systems as biometric security access units, dual fire control systems and Uninterruptible Power System (UPS). These systems provide a smooth and seamless transition to the recovery scenarios instituted at NITC. Three diesel powered backup generators assure NITC customers their data processing will continue uninterrupted in the event of a power outage.

Shifts and Time Information

The NITC computer environment is operational 24 hours a day, 7 days a week, with the exception of a scheduled maintenance.

A current weekly operations schedule may be obtained by executing Clist **KCHELP** from ISPF option 6 or the READY or by typing NEWS at the Teleview Main Menu command prompt.

Job Processing Standards

NITC job processing standards include job class, job turnaround, and deferred processing standards. The following table displays class categories, job classes and CPU time limits for processing, associated to available devices:

Job Class Requirements Table

CLASS CATEGORY	CLASS	CPU TIME LIMIT (Minutes)	MOUNTABLE DEVICES/STEP NUMBER
NON-SETUP	**K**	**1**	**0**
(No Mounts)	**C**	**5**	**0**
	D	**1439**	**0**
SETUP	**F**	**10**	**1**
(Mountable)	**G**	**1439**	**2**
	B	**5**	**6**
	I	**1439**	**6**
	N	**1439**	**12**
TAPE MOUNTS:			
RD Tapes	**L**	**60**	**2**
WSC Tapes	**6**	**1439**	**12**
CA-IDMS:			
Production DC	**T**	**1439**	**0**
Test DC	**U**	**1439**	**0**
CV Production	**O**	**1439**	**2**
CV Test	**Q**	**60**	**2**
Journal/Oracle Archive Offloads (*)	**J**	**10**	**1**
ORACLE Instance	**M**	**1439**	**0**
FOCUS SU	**W**	**1439**	**0**
FOCUS BATCH/SU	**5**	**80**	**1**
PRINT	**P**	**10**	**2**
Special Handling	**H**	**1439**	**12**
CICS	**R**	**1439**	**4**
CICS	**0**	**1439**	**4**
FOCNET	**7**	**1439**	**0**
TRACS	**8**	**1439**	**0**

(*) = Single step and PGM=IDMSBCF for CA-IDMS or PGM=ARCHIVE for Oracle.

NOTE: The number of mountable devices should include those used in Tape Sorts.

Customers requiring additional processing requirements, unique job classes or additional initiators should contact the SNCC at (816) 926-6681.

Job Class Table

Job Classes	
No setup classes	**C**, **D**, **K -** Designed for jobs that require no mountable devices.
Setup classes	**F** - used for jobs requiring only a single mountable device. **G -** permits up to two mountable devices per step. **B, I** - permit several mountable devices (6 maximum) per step. **N -** permit more mountable devices (12 maximum) per step.
Special handling for large jobs	**H** class is for large jobs. Because jobs in this class can impact system efficiency, scheduling and running in this class must be carefully controlled. Class H jobs are placed on HOLD and released by Computer Operations as resources become available. At customer request, SNCC will estimate when class H jobs will run.
Special Reserved Job Classes	**O** - CA-IDMS/CV production **Q** – CA-IDMS/DC test **T** – CA-IDMS/DC production transaction processing **U** – CA-IDMS/DC test transaction processing **M** – Oracle Instances **J** – CA-IDMS Journal and Oracle Archive Offloads **P** – Edit processes that spool data to SYSOUT . **W** – FOCUS SU **5** – FOCUS Batch SU **7** – FOCNET

Priority Scheme

There are four levels of priority within each class. Cost factors are expressed as a multiple of the base rate:

- Priority 13 - The highest priority, cost factor of 3
- Priority 12 - Intermediate priority, cost factor of 2
- Priority 11 - Normal jobs
  Prime time, cost factor of 1
  Non-prime time, cost factor of .60
- Priority 2 - Deferred
  Prime time, cost factor of 1
  Non-prime time, cost factor of .40

Prime time has been defined as 0800 through 1600 hours Monday through Friday,

excluding holidays.

The priority level desired will be designated by specifying the priority value on the job card in the form PRTY=XX, where XX is a one or two digit number (i.e., no leading 0's). This field becomes another keyword parameter on the job card.

By default, if the Priority level parameter is omitted from the job card, "PRTY=11" will be used.

In this example intermediate priority is desired:

```
//JOBNAME JOB  (123456789012,DIST,#),'USER INFORMATION',
//      CLASS=C,TIME=(,20),MSGLEVEL=1,PRTY=12,MSGCLASS=A
```

All jobs submitted with PRTY=2 and a Job class of either A, B, C, D, F, G, I, K, N or 6 will be placed on HOLD and manually released by the NITC operators after 1600 hours CT, during the non-prime shift. The NITC offers no guarantee that these jobs will be processed the same day they are submitted. They will be released at the discretion of the NITC operators.

Rate Differentials Table

Job PRTY	Service Request	Prime Time Surcharge	Prime Time Differential	Non-Prime Time Surcharge	Non-Prime Time Differential
13	High Priority	3.00	2.33	3.00	2.33
12	Intermediate Prty	2.00	1.33	2.00	1.33
11	Normal Charge	1.00	.90	.60	.50
2	Deferred-Normal	1.00	1.00	.40	.40

Priority surcharges apply to computer charges. When turnaround standards are not satisfied, the lower differential will be used. Turnaround standards DO NOT apply to Priority 2.

Job Turnaround Standards Table

Priority:	11	12	13
CLASS	TIME	TIME	TIME
B	1.5 hr	1.0 hr	30 min
C	45 min	30 min	20 min
D	3.0 hr	2.0 hr	1.0 hr
E	0.5 hr	0.5 hr	0.5 hr
F	4.0 hr	3.0 hr	1.0 hr
G	8.0 hr	5.0 hr	2.0 hr
I	8.0 hr	5.0 hr	2.0 hr
J	45 min	20 min	10 min
K	15 min	10 min	5 min
N	10.0 hr	8.0 hr	6.0 hr
O	1.0 hr	45 min	30 min
Q	1.0 hr	45 min	30 min
H	Special	Handling	

Time, for a job, is measured from read in to initiation time. If a job has the same jobname, as another job awaiting execution or executing, the job turnaround standards do not apply.

JES2 Print Priority

The priority of the output generated, by jobs, at the priority 13, 12, or 11 levels will be the same as the job card priority. For all other levels, the default is PRTY=11.

The default limit or SYSOUT lines produced per job is 500,000 lines. Use the estimated print line count in the accounting field or on the JOB card or on the /*JOBPARM card to override this default value. A $HASP375 message is issued when the output estimate is exceeded. The job is cancelled, without a dump, when the job exceeds the estimated number of lines.

Care should be exercised when creating large SYSOUT since space on the JES2 spool can be exhausted. If more than 500,000 lines are expected, the output should be split-out to several tapes and spooled to SYSOUT as separate jobs. Extremely large jobs (10 or more tapes) should be coordinated through the System Network Control Center (SNCC), (816) 926-6660.

RJE Route Print Local Output

NITC remote job entry customers, who have issued a route print local command through RJE to direct output to the system's central printers, must make arrangements to pick up their materials. Materials not having a normal output bin may be picked up in the System Network Control Center (SNCC), (816) 926-6660.

Daily Queue Maintenance

Daily maintenance of queues will be executed as follows:

- Purge READY output that is 15+ days old.
- Purge READY output in outclasses D, F, and N that is 4+ days old.
- Purge HELD output that is 8+ days old.
- Purge HELD output in outclasses D, F, and N that are 4+ days old.
- Purge all jobs waiting execution 10+ days old.
- Purge all jobs waiting JCL conversion 7+ days old.
- Purge all jobs waiting NJE transmission 7+ days old.
- Purge all HELD jobs waiting the output process 8+ days old.
- Purge all READY jobs waiting the PRT/PUN process 15+ days old.

SYSOUT

The following table describes; SYSOUT classes available, there association to NITC LOCAL and user remote printing destinations, along with time limits for print queue purge:

SYSOUT Job Class Table

Class Type	Class	Description	Print location/Dest	Purged After
Print (Not Held)	A	Standard print class	Local, RMT, VPS	15 days
	C	Special Forms	Local, RMT, VPS	15 days
	E	WSC Xerox special (4090)	WSC in WDC	15 days
	G	SAS Graphic on VPS	VPS dest	15 days
	H	FSA Xerox Standard print	FSA-Beacon St., K.C. /KCMO	15 days
	I	FSA Xerox Special	FSA-Beacon St., K.C. /KCMO	15 days
	J	RD RMT Print (RMT Download)	RD /RMTxx	15 days
	L	SYSLOG offload/archived	Offloaded-no print	15 days
	M	WSC Microfiche	Offloaded-no print	Manual
	P	RD Punched Paper	RD-St. Louis, MO /STL8	15 days
	T	TRMS Accumulator	TRMS-no print /TRMSxx	Immediate
	X	RD St. Louis, MO	RD-St Louis, MO /STL8	15 days
	1	WSC Standard print class	WSC in WDC	15 days
	2	WSC Special Forms	WSC in WDC	15 days
	3	$AVRS – Written to $AVRS databases	$AVRS-no print /$SAVRxx	Immediate
Punch	B	Standard punch class		15 days
	K	Standard punch class (RD RMT download)	RD RMTs	15 days
Hold	D	Purged after 4 days (dumps, STC, etc)	None	4 days
	Q	Purged after 8 days (hold class at NFC)	None	8 days
	R	Purged after 8 days	Local	8 days
	Y	Purged after 8 days (hold class for FAA)	None	8 days
Cond Purge		Output is purged on normal termination.		
	F	Output is HELD if abnormal termination (JCL Error, Abends, Cond= exceeded on Jobcard)		4 days
		Out is purged on normal term.		
	N	Output is print ready if abnormal termination (see class F)	Specify route/dest (Forest Service uses)	4 days
	All Other	Purged immediately		

Batch procedures are available for WSC users receiving SYSOUT on the Xerox 4090 LPS. Include one of the following steps before the applicable production step to receive desired output.

1. Landscape/Duplex mode, 132 columns and 66 lines, both sides:
   //STEP0 EXEC DUPLEX (DEFAULT)

2. Landscape/Simplex mode, 132 columns and 66 lines, one side:
   //STEP0 EXEC SIMPLEX

3. Portrait/Duplex mode, 90 columns and 60 lines, both sides:
   //STEP0 EXEC PORTDUPX

4. Portrait/Simplex mode, 90 columns and 60 lines, one side:
   //STEP0 EXEC PORTSIMX

5. Portrait/TWO UP Simplex mode:
   //STEP0 EXEC PORT2UP

6. Portrait/TWO UP Duplex mode:
   //STEP0 EXEC PORT2UPD

Job Failures

Contact (816) 926-6681 regarding jobs that abnormally terminate due to NITC software or equipment problems.

Job Validation Errors

The following messages are issued when a job encounters a standard violation detected by the NITC job validation routine:

NITC01:	Reserved for future use.
NITC02:	Too many mountable devices per step. Check number of tapes used per step in JCL and use the appropriate class. See Job Class Requirements for Class specification.
NITC03:	Setting region size equal to 0Kis not allowed.
NITC03A:	Not Used.
NITC03B:	Region below the line must be 1M thru 9M.
NITC04:	Time limit too large for your class. Check time parameter in JCL and make sure that it corresponds with the time limit allowed for the class specified. See Job Class Requirements for time limits allowed by class.
NITC05:	Job class invalid. Make sure the class specified in the JCL is a valid Job class.
NITC06:	Reserved for future use.
NITC07:	PRTY=2 & Illegal Class - Set PRTY=11. The class selected was

not eligible for deferred processing. The priority was changed to PRTY=11.

NITC08: Illegal Priority - Set PRTY=11 an illegal priority was used. The job priority was reset to PRTY=11. See Priority Scheme.

NITC09: Illegal Job name for account code. Make sure that a valid job name prefix for your agency was used in the job name. Check with your Security Officer

NITC10: No ADDRSPC=REAL allowed. Check JCL for V=R. Virtual Storage= Real is not allowed.

NITC11: Illegal logonid for account code. The LOGONID used with the job does not have authority to use the account code specified in the job. Check with your Security Officer.

NITC12: Invalid account code. The account code used with the job is not a valid account code. Check with your Security Officer.

NITC13: Illegal status code. This job card parameter is optional but if used it must be numeric. See the Job Definition and Requirements chapter of this handbook for an explanation of Status Code.

NITC14: Illegal DASD unit name. An invalid unit name has been specified. NITC DASD unit names are SYSDA, STORE, and LARGE. (LARGE is to be used when the data set is 200 cylinders or bigger). See the Data On-line Storage Facilities chapter of this handbook for more information.

NITC15: Use of binding agent is not authorized. Check to see you are using a valid binding agent.

NITC16: Reserved for future use.

NITC80: TYPRUN=SCAN does not check account code or JOBNAME. This is a message issued for every job run with a TYPRUN=SCAN informing the user that account code, LOGONID vs. account code, nor account code vs. job name validations are not executed for this type of jobs. See Job Definition and Requirements of this handbook for an explanation of the TYPRUN=SCAN parameter.

DTMNITC81: Unauthorized use of TIME=1439 parameter.

DTMNITC90: Job has an improper class. Where possible the class is changed to a valid job class. See CLASS= in the message to identify the new job class and see the table of valid job classes in this Handbook.

Chapter 4. Information/Management (INFO/MAN)

Introduction

IBM's Information Management (INFO/MAN) software is used as a tool to assist NITC employees in problem and change management. This online system provides a means for recording, retrieving, and managing information about site problems, changes, and system configurations on a daily basis.

NITC customers may enter "problem" records into the system or make requests, ask questions, or make suggestions. Customers who have been given access by their agencies may view problem records in the database and display or print reports on problems being addressed.

The primary function of INFO/MAN is reporting and tracking hardware and software problems. It is not a vehicle to discuss policy issues. These issues must still be addressed through the Office of the Director.

Once a problem record has been entered, NITC personnel are responsible for assigning resources to the problem, updating the record with information as it becomes available, documenting the problem resolution and closing the problem record upon resolution.

INFO/MAN runs under ISPF and requires a 3270-type terminal with a screen size of 24 lines by 80 characters. It is similar to TSO/E in that its operation relies heavily upon native commands, numeric screen options, and PF keys. Some data elements entered are validated and others, like the description field, are freeform.

Since INFO/MAN is based on a structure of interactive dialogs, customers should be careful that the session does not become too nested or results will prove unpredictable.

The remainder of this chapter addresses some basic information regarding using INFO/MAN. An online tutorial is also available within INFO/MAN.

Helpful Commands

Following are short descriptions of the most common native INFO/MAN commands. Most commands may be shortened to the first two letters. For example, EXECUTE may be entered as EX.

QUIT	Exit INFO/MAN
CANCEL	Discontinue current dialog; disregard previously initiated action; do not save

END	Complete this panel, return to other options, and save data (PF3)
INITIALIZE	Discontinue all dialogs and restart back at the Main Menu (PF6)
BACK	Return to previous panel
RECALL	Redisplay the last data item or command entered (PF4)
HELP	Help for the product software (PF1)
GLOSSARY	Browse of values present in database
SEARCH	For search arguments to locate records within the database
EXECUTE	Invokes a pre-programmed dialog for ease of data entry
ISPF KEYS	Display the default PF key values. Users may change these defaults, add search arguments or other commands.
DISPLAY	Display individual problem record information
PRINT	Print problem record information
ISPF PRINT	Screen print data
PRINT ALL	Prints a search results list
SE =0 KEYW	Format for conducting keyword searches

Sign on and User Profile

1. To utilize INFO/MAN, you must possess a TSO logon id. In addition, your logon must be included in your agency's privilege class defined within the INFO/MAN product. If you need to be added to your agencies privilege class, contact the personnel at your agency who approve INFO/MAN access requests.

2. Sign on to **TSOB.**

3. At the ISPF Primary Option menu➔ **=I**

4. Select option 2 from the INFO/MAN Primary Options panel if you wish to update your user profile defaults.

   Entering a value into a field may be accomplished in a variety of ways:

   - Tab down to the element to be entered and type the value.

   - Enter the number of the element and the value for it on the command line, separated by a comma. Example 1,A.

     (You may also string several data items together to fill more than one field. Example 1,A,2,003,4,U001 enters an “A” into field 1, “003” into field 2 and “U001” into field 4.

   - Enter an equal sign (=) into a field. Whatever is designated the default for that field in your user profile will be entered into the field.

   - Entering a null symbol (^) will set a field to no value (eliminate spacing or zero out).

Changing the SYSOUT class to “A” and leaving the destination printer blank will cause your INFO/MAN print outs to be routed to a NITC printer.

5. Press **PF3** (or type END) until you return to the Profile Summary panel.

4. Selection option **7** from the Profile Summary panel to update user profile name, department and phone number defaults.

5. Press **PF3** (or type END) to return to the Profile Summary panel.

6. Select option **10** (Permanent Profile End) to permanently save your profile changes, otherwise, your changes will only be used for your current INFO/MAN session.

Entering Problem Records

1. From the Primary Options panel, type **EXECUTE CUSTOMER.**

2. The problem reporter entry screen will be prefilled with the default information from your user profile, as well as current date and time. Any of these fields may be changed.

3. The following are mandatory fields and must be entered:

   - Initial Severity
   - System name
   - Description (should be concise)

   To see valid values for a field, key the item number on the command line and press enter.

3. Press **PF3** (or type END).

4. Next a description panel will display. Enter description of the problem. When complete, press **PF3** (or type END). The message "SRC CUSTOMER HAS ENDED" will display.

5. Press **PF3** (or type END) to save the record you entered or type **CANCEL** if you do not wish to save the record. If you have successfully entered the INFO/MAN record, an INFO/MAN record number assigned to that problem in the INFO/MAN database will appear. Note the record number for future reference.

Accessing INFO/MAN Records

1. If the record number for a particular INFO/MAN problem is known, use the following command to view the record. The following command will display INFO/MAN record 5678:

DI R 5678

The record will be displayed and the customer may view the status (selecting option 2) of the record and find who has been assigned to the problem. The free form text and notes entered for the record may be viewed by selecting option 3.

2. A record may be printed without displaying it by issuing the print command:

   PRI R5678

3. You may also search the INFO/MAN database for specific records by using the SEARCH command. This command will produce an online search results list that will show the record id, date entered, priority, assignee, and short description for every record in the database matching the search criteria.

   To search the database type the **SEARCH** command followed by the PREFIX for the field being searched followed by a "/" and then the value you are searching for followed by a period. (The PREFIX for the field can be found the upper right hand corner of the help screen for the field when doing an update or entering a new record.)

   INFO/MAN also allows logical operators for compound searches by using the vertical bar (|).

   Sample searches:

   SE PERS/JOHNSON. - Lists records reported by last name JOHNSON.
   SE PERC/JOHNSON. - Lists records being tracked by JOHNSON.
   SE PERA/JOHNSON. - Lists records assigned to JOHNSON.
   SE PER*/JOHNSON. – Lists all records reported, tracked and assigned to JOHNSON.

   SE STAC/INITIAL - Lists all records in INITIAL Status (not yet opened and assigned).

   SE COBOL - lists all records in the database with COBOL in the description.

   SE DAT*/2001/01/01 - lists records where any action occurred on 1/1/2001 (date entered, date occurred, date assigned, date opened, date closed, etc…)

   SE COMK/CA - lists all records with CA in the Key Item Affected field. This would return all records with CA-7, CA11, CA/11, CA-11, and CA-IDMS in the Key Item Affected field.

SE IDMS | DB2 - is a compound search that displays records which contain either IDMS or DB2 in the description field keyword in the description

SE FOCUS STAC/OPEN - Displays all open items with FOCUS in the description

SE ^STAC/CLOSED - ^ Symbol is used with the prefix. This search displays records with a status other than "CLOSED".

SE INFO. STAC/INIT. ^PERA/. DATO/2000/12/01 - 2000/12/31 GROS/F. | GROS/A.

- Lists records the meet the following criteria:

INFO in the description (INFOMAN, INFORMATION, ETC.)
in INITIAL status (that is unassigned)
Occurring between 12/1/2003 and 12/31/2003
the name of the reporter's Agency (PREFIX:GROS)
begins with either an F or A

Prefixes

Below is a list of the most commonly used prefixes in INFO/MAN. For more information on the types of entries present in the INFO/MAN database, select 8 (GLOSSARY) from the Primary Options panel or type the command GLOSSARY. By leaving the option blank and pressing enter, you can scan the entire glossary of data element and keyword values, as well as their number of occurrences. Prefixes are also listed in the online tutorial.

CLAE - Entry privilege class PERA -Person assigned
COMK - Key item affected PERC -Tracker
DATA - Data assigned PERS -Problem reporter
DATE - Date entered PH -All phones
DATO - Date occurred PRIO -Current priority
DATR - Date closed RNID -Problem number
DATX - Date opened STAC -Problem status
GROA - Assignee department TIME -Time entered
GROC - Tracker department TIMO -Time occurred
GROS - Reporter department TYPE -Problem type

Chapter 1. Televiewv

Introduction

Televiewv is an advanced session manager from Computer Associates that is the recommended method for connecting to NITC systems. Televiewv allows a user to establish multiple concurrent online sessions. Since the user may switch between these sessions with minimal effort, the need to disconnect from one online application before establishing another is eliminated.

Security

Televiewv provides additional system security through its interface to ACF2. The Televiewv menu panel can be configured to display only those applications for which a user is authorized and will indicate the application's availability. Televiewv can be tailored to provide automatic log on to any application that incorporates ACF2 security.

Terms

Televiewv Profile - A Televiewv profile determines what applications a user sees displayed on the Televiewv main menu panel.

Scripting - Scripts provide automatic logon to applications. Scripting enables a user to bypass normal sign-on activities.

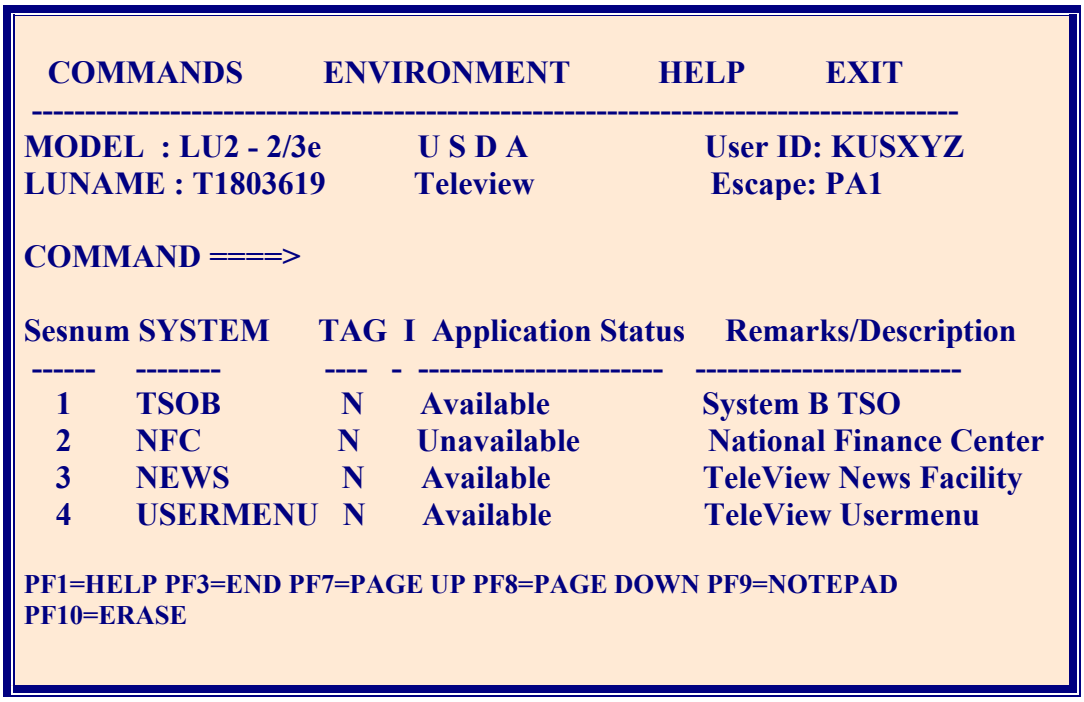
Usage

Your PC is probably configured to invoke Televiewv when you click on the icon to connect to NITC. If not, Televiewv may be invoked by typing TELEVIEW anywhere an applid may be entered.

At the first Televiewv screen panel, the user is prompted for an ACF2 user ID and password. This is your TSO id and password. If the password has expired, a message prompts the user to enter a new password. Once the password has been validated, a Televiewv main menu panel is displayed.

If you only see options NEWS and HELP, your agency security staff has not assigned a Televiewv profile for your user ID. To request a Televiewv profile containing the applications you need to access, contact your agency security officer.

If a profile has been established for you, the following is an example of what the main menu panel might look like. The applications you actually see will depend upon the profile requested by your security officer.



Action Bar Area

The first line of the main menu panel is the action bar. It contains four action options you may select by placing the cursor to the left of the item and pressing ENTER. The following features of Teleview are described only briefly. See the Teleview User Guide for more information. See the References section of this chapter for information on available Teleview manuals.

COMMANDS returns a pull-down menu that lists the Teleview commands available to you. (Consult the Teleview User's Guide for an explanation of these commands.)

ENVIRONMENT returns a pull-down menu that provides the region id, date/time, escape key setting, the character used for Trigger switching, and a subsequent pull-down for remarks.

The Escape key is the attention key used for session switching. You may change the escape key on this pull-down by typing in a valid escape key (ATTN, PA1, PA2, PA3, or CLEAR). Changing the Escape key. (Making any update in Usermenu will save the escape key change permanently.)

The CMDCHR field on this menu identifies the character used for Trigger Switching. The default character is a period (.). You may change this by typing in another

character that you wish to use.

HELP returns a pull-down menu to give you help on the use of the main menu. (General help on Televiewis available by pressing PF1).

EXIT returns a pull-down menu for various ways to logoff Televiewi. Place the cursor to the left of your LOGOFF choice and press ENTER.

Note: LOGOFF DISC command is not applicable to this environment and has been disabled.

Information Area

The information area on the main menu is the area directly below the action bar. Televiewi error messages are displayed on the first line of the information area. Your terminal model, terminal id (luname), user ID and escape key designated for session switching are displayed in the remaining information area.

Command Area

The command area on the main menu is located below the information area where you may specify a Televiewi command or session number.

Application Section

The application section displays the list of applications defined by your Televiewi profile. The "Tag Key" column displays the Tag key designated for one key session switching. The Application Status field indicates whether an application is available, unavailable, active or other status messages.

PF Keys

PF1 Televiewi Help.
PF3 If no pull-down menus are active, PF3 will issue a LOGOFF command.
If a pull-down is active on your screen, PF3 will back out one level.
PF7 Page Up one screen in the Application Section.
PF8 Page Down one screen in the Application Section.
PF9 Select this option to add a Note or Reminder.
PF10 Delete all active Notes and Reminders.

Establishing Sessions

Sessions are established in any of the following four ways. For example, on our sample menu, TSOB is option 1. You could log onto TSOB using one of the following methods:

1. Move the cursor to the left of the TSO session option and press ENTER.

2. Type "TSOB" on the prompt line and press ENTER.

3. Type session number 1 on the prompt line and press ENTER.

4. Press ATTN then PF1.

If the connection to TSOB fails, an error message will display. If the connection is successful, Teleview will switch the terminal to TSO.

Because you have already entered your TSO user ID and password upon entering Teleview, Teleview will pass them to TSO automatically for you.

To override the ACF2 lid and password passed automatically to the application, at the main menu panel prompt line, type "n (user ID)" (where n equals the option number of the TSO application) and press ENTER.

This applies to only to TSO applications. For CICS applications using an ACF2 interface, you can log on to the application with a different user ID by turning the scripting off. This is accomplished by typing CMLOFF at the Teleview main menu panel.

To exit an application with out disconnecting and return to the Teleview main menu panel, press the assigned ESCAPE key press ENTER.

To establish another session, at the Teleview main menu panel, select another option.

Session Switching

There are three alternatives available for session switching. For all three examples, assume you have already established two sessions, TSOB (option 1) and NFC (option 2) and you are currently viewing your TSOB session.

1. The first method is "Escape Switching". This method is the most commonly used and requires only two keystrokes.

   Press the PA1 key to put the current session "on hold". Next, press PF2 to switch to your NFC session.

2. The second method is "Tag Key Switching". This method allows a session switch with one keystroke. "Tag Keys" are designated via Usermenu. See the Teleview User Guide for information on setting Tag keys.

The third method is "Trigger Switching". This method allows you to enter a session name, the leading characters of a session name, or a session number from another session. A character designated as the session switching trigger must precede the name or number you enter. The default trigger character is the period (.).

Logging Off

The LOGOFF command is entered at the Teleview main menu prompt line or selected from the EXIT pull-down menu. There are several options available for logging off Teleview:

1. If you have no active sessions, enter or select **LOGOFF**. This command is not permitted if the terminal has "Active" (bound) sessions.

2. To logoff just one of your sessions, type "**LOGOFF n**" where n is the number of the session you want logged off.

3. To logoff Teleview and all active sessions, enter or select "**LOGOFF ALL**".

It is not necessary to logoff within the application, although you can. Teleview will disconnect your sessions for you with the "LOGOFF n" or "LOGOFF ALL" commands.

It is important to note that Teleview will not clean up your sessions for you. If you are editing a file, for instance, you must first save your file. A "LOGOFF ALL" or "LOGOFF n" will terminate the session without saving your changes.

Security

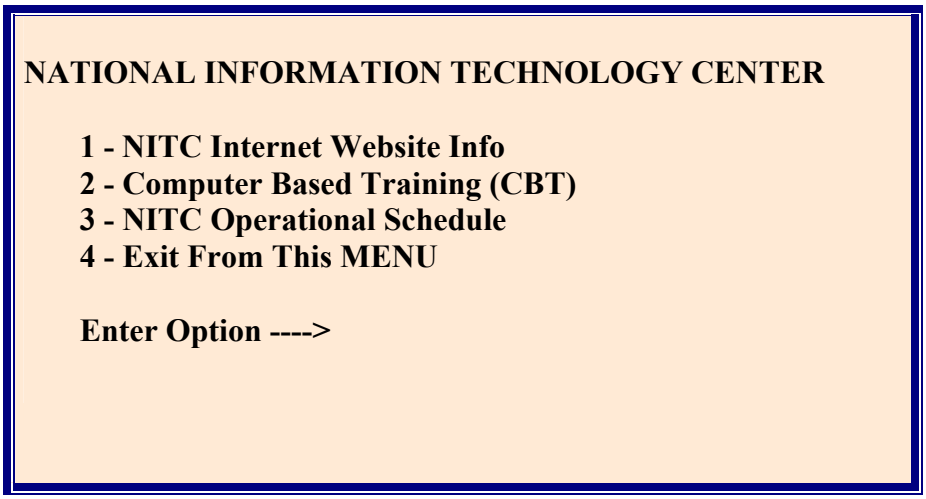
If you wish to leave your terminal while you have active sessions, you may "lock" your Teleview session. To do this, at the Teleview main menu panel, enter "**LOCK**" or select the LOCK command from the COMMANDS pull-down menu. To "unlock" your terminal, enter your ACF2 password.

For enhanced security, NITC has tailored Teleview to provide an "automatic lock" feature. A user's keyboard will lock after 15 minutes of inactivity, requiring entry of the ACF2 password to "unlock" the terminal.

Teleview NEWS

The Teleview NEWS option is similar to a system-wide bulletin board that is maintained by the NITC. To access Teleview NEWS, type **NEWS** at the Teleview Main Menu panel.

Teleview NEWS has the following information of interest to NITC customers:



1 NITC Internet Website Info – lists NITC websites.

2 Computer Based Training – provides a list of available courses and information on each course to aid customers in deciding what courses they would like to take and what prerequisites may be needed. For more information, see the Computer Based Training chapter of this Handbook.

Information is also provided on how to register for a CBT course and how to sign on and take a course.

3 NITC Operational Schedule – provides a weekly report on IBM systems availability.

4 NITC Point of Contact List – provides information on support personnel for systems services, hardware and software.

Usermenu

USERMENU is a facility that allows individual Teleview users to "customize" their Teleview main menu without affecting other users who share the same profile. Users can select which of their applications will appear, and in what order. With USERMENU a user can delete or update applications on their main menu. A user can also change data on the main menu.

Before using USERMENU, please be aware of the following:

Once you exit USERMENU, your changes are applied to the main menu automatically.

If a new application is added to your shared profile and you are using your own USERMENU defined menu, you will NOT see that the application has been added. This also applies if the menu name has changed. The application will show up as a deleted application in USERMENU. If you wish to add it to your menu, you must use the

USERMENU "Add Procedure" or you may wish to restore your profile back to its original configuration.

To restore your profile back to its original configuration, go into USERMENU, press PF2 twice and press PF3 to exit USERMENU. Log completely off Teleview and back on.

To access USERMENU, it must be an application option on your Teleview main menu. If you wish to have USERMENU added to your profile, contact your IM Security Officer. For more information on using USERMENU, consult the Teleview User Manual.

Policy

Formal requests for Teleview access, profile development or changes to an existing profile must be routed from an agency security officer to the Center security staff.

Requesting a Profile

When requesting a new Teleview profile, be prepared to provide the following information to your IT Security Officer:

1. The VTAM applids of the applications you wish to access. List the applications in the order to be displayed on the Teleview main menu panel.

2. Provide a menu name for each application if different from the actual VTAM applid.

3. List an option number for each corresponding application.

References

Teleview manuals may be viewed in BookManager. For information on how to access and use BookManager, see the BookManager chapter of this handbook.

The Teleview User Guide may be ordered from Computer Associates by contacting:

Computer Associates
1 Computer Associates Plaza
Islander, NY 11788-7000

Attn: Documentation

or call 1-800-841-8743

Addition reference information may also be obtained by accessing url: http://www.ca.com/ and entering in the SEARCH box: **Teleview**.

Chapter 2. Time Sharing Services (TSO/E)

Introduction

The NITC provides USDA customers interactive timesharing services through IBM's Time Sharing Option (TSO/E). This chapter is intended to provide customers with a general overview of the timesharing services NITC offers and information for connecting and using these services.

TSO/E allows a customer to have "conversation" with the computer. TSO/E commands are entered through a terminal and the computer responds with the appropriate action and output. TSO/E makes it easier for both the programmer and non-programmer to use mainframe computer services.

Using TSO/E, customers may create on-line files through a terminal or retrieve existing on-line files. Once a file is accessed, it can be manipulated, altered, saved or executed.

Terminal Requirements

NITC supports most 3270 compatible terminals.

Logon Requirements

You must have a valid ACF2 logon id to access TSO/E. If you do not have a TSO logon id and password, contact you Agency Security Officer to establish one for you.

Depending on the parameters set for your logon id, you may be prompted for logon procedure (or logon proc) when you log on to TSO/E. The proc you use will determine what files are allocated for your TSO/E session.

NITC maintains logon proc $SPF which will allocate the files necessary to execute TSO/E and ISPF/PDF. In addition, this proc will allocate the data sets necessary for accessing other software products NITC offers via TSO. These products may be accessed by typing **=D** from an ISPF panel and then selecting the desired product.

Starting a TSO Session

Check with your agency technical support personnel on what methods they have in place for you to access TSO. Once you have accessed TSO, your logon id will be validated by the system. You will see the message: (USER ID) LOGON IN PROGRESS AT (TIME) ON (DATE).

Depending on how your logon id was configured, upon signing on to TSO, you will be at either the READY prompt or the ISPF Main Menu Panel. If you are at the READY prompt and wish to access ISPF, type **%ISPF**. At the READY or in ISPF, the system is now ready for you to enter a command.

Terminating a TSO Session

To terminate your TSO sessions, type LOGOFF at the READY prompt or **=D.X** at the ISPF Main Menu panel.

Session Drops

Due to a telecommunication, LAN, or system problem, a terminal may occasionally prematurely terminate your session. If you were editing a data set when this occurred, TSO will attempt to recover the file you were editing the next time you attempt to use the ISPF Editor (provided your ISPF Edit Profile had recovery set to on).

You can check the recovery setting in your ISPF Edit Profile a variety of ways:

1. Type **PROFILE** at the READY prompt.
2. Type **TSO PROFILE** from any ISPF panel command line.
3. Type **TSO PROFILE** from ISPF Option 6.

Adding RECOVER to the PROFILE command (Ex: PROFILE RECOVER) will specify that the edit recovery function be used for all ISPF Edit sessions. You may also set recovery on for a specific data set by typing RECOVERY ON while editing the data set in the ISPF Editor.

Advantages of TSO/E

- On-line creation and editing of program and data files.
- Processing programs with interactive communication between customer and program.
- Simplified JCL and program submission.
- Simplified routing of output.
- Rapid, convenient turnaround.
- Customer control of computer usage without operator intervention.

Broadcast Notices

NITC uses broadcast notices to communicate important information to its customers. While these notices may cover a wide spectrum of topics, they are specifically intended to advise customers of changes in NITC services.

If a broadcast message has been issued, it will display on your terminal immediately

following a LOGON. No action on your part is necessary to retrieve the message.

In an emergency situation, NITC may opt to transmit a message to terminal customers during their active session. These notices are of particular importance and should be heeded.

TSO/E Help

TSO/E has built-in features designed to assist customers. One method is called Prompting Dialogues. When entering a command at ISPF option 6 or the READY prompt and you cannot recall the parameters of the command you want to enter, TSO/E will prompt you for the parameter.

For instance, if you type **LISTDS** to list a data set and did not provide the data set name in the LISTDS command, TSO/E will prompt you for the data set name. If you do not understand the prompt, enter a question mark (?) and press return to retrieve an explanation of the message. If additional information is not available, TSO/E will respond with no information available.

Another method is the HELP command. The HELP command provides customers with information for all TSO/E commands. At the most general level, customers can enter **HELP** and receive a list of all commands and a brief explanation of their functions.

Customers can retrieve extended information about a specific command by entering HELP followed by the command name, for instance, **HELP LISTDS**.

Customers can also limit retrievals to the syntax operand requirements of a selected command, for instance, **HELP LINK SYNTAX HELP LINK OPERANDS**

Information can be retrieved about a specific operand of a command by enclosing the name of the operand in parentheses immediately after the operands keyword, e.g. **HELP LINK OPERANDS(XCAL)**

Function Keys

Program function keys are intended to perform a number of full-screen functions and can reduce the repetitive effort of keying commonly used terminal commands. In full screen mode the following functions are assigned to PF keys 1 thru 12:

PF1 HELP
PF2 SPLIT SCREEN
PF3 END
PF4 RETURN
PF5 REPEAT FIND
PF6 REPEAT CHANGE
PF7 SCROLL UP
PF8 SCROLL DOWN

PF9 SWAP SCREENS (for split screens)
PF10 SCROLL LEFT
PF11 SCROLL RIGHT
PF12 NEXT 1

Limited modification of assigned function keys is permitted but generally consists of exchanging key assignments. This can be accomplished by selecting option **0** (Settings) from the ISPF Main Menu panel. Option 0 will also allow you to set other terminal user parameters.

TSO/E Superset Utilities

For information on TSO/E Superset Utilities, see the TSO/E Superset chapter in this Handbook.

Training

Online training is also available for TSO. See the Computer Based Training chapter of this Handbook for more information on available CBT training.

References

For more information on TSO/E, access the TSO/E bookshelf in BookManager. For help with using BookManager, see the BookManager chapter of this Handbook.

Chapter 3. CICS Transaction Server

Introduction

Customer Information Control System (CICS) is a general-purpose data communication or on-line transaction processing (OLTP) system. Utilizing an online system controller and some batch utilities, CICS is capable of supporting a network of many thousands of terminals. CICS is a specialized operating system that provides an environment for the execution of on-line application programs. CICS supports interfaces to files and a variety of telecommunications subsystems.

CICS Transaction Server integrates CICS with a set of other supporting software. New features of CICS Transaction Server include extended Parallel Sysplex support, improved system management, application support, and solution ennoblement for network computing.

Functions

CICS acts as a supervisor, providing such services as file control, program control, storage management, buffer management, functional recovery, mapping services, and transaction back out.

CICS provides:

- Operating system services.
- Functions that application programs need for communication with remote and local terminals and subsystems.
- Control of concurrently running programs serving many on-line customers.
- High availability and faster recovery.
- Controlled access for authorized customers (by means of internal CICS security features and, where appropriate, external security management software) to databases and files, using various database products and data access methods.
- Communications with other CICS systems and database systems, both in the same computer and in connected (remote) computer systems.
- Many other features essential to the creation, operation, and upkeep of a

secure online system.

CICS Transaction Server provides:

- Standard web browser support through the HTTP protocol.
- Direct network connection with an optional interface through WebSphere Application Server.
- Powerful tools for web-enabling BMS applications.
- Access to CICS 3270-interface applications from the web without changing existing code through a 3270 bridge.
- Ability to modify the screen layout presented to web browsers to exploit the features of HTML and JavaScript.
- Visual Age for Java support to enable Java application programs to run under CICS control.
- Enhanced security for internet use, such as, SSL support.
- An MVS Logger that lets MVS handle all CICS logging.

Data Communications

CICS terminal control and basic mapping support (BMS) facilities make it easy to write application programs that use terminals of different types. With BMS you can use the same application program with different terminals, while using the full range of facilities that each terminal provides. The CICS inter-communication facilities allow two or more systems or regions to communicate and/or share terminals and other resources.

CICS inter-communication offers:

- Distributed transaction processing, which allows communication between transactions executing in different systems.
- Asynchronous processing, which allows a CICS transaction to start a transaction in a remote system and pass data to it.
- Transaction routing, which allows operators of terminals in one CICS region to run transactions in any connected CICS region.

Data Handling

Interfaces within CICS allow access to data held in a variety of places including:

- Databases
- Standard operating system data sets

1. Databases - give the greatest degree of data independence. Both batch programs and CICS transactions can share data with equal freedom for either access or update, and with full database integrity.

2. Standard data sets - offer a limited form of database facility, since information is organized in individual files. These are processed by CICS file control which interfaces with the appropriate access methods including:

   Virtual Storage Access Method (VSAM)
   Basic Direct Access Method (BDAM)

Language Compilers and Assemblers

CICS language compilers and assemblers supported by NITC are:

IBM COBOL
Assembler H

NOTE: A waiver must be obtained from the NITC security officer to use the system assembler.

Policy

CICS is available for all Center customer agencies. An agency desiring to use CICS as a communication server is requested to specify a CICS administrator to serve as point of contact with NITC for CICS implementation and related issues.

The cost of a NITC-maintained region is prorated based on the CPU usage of each agency. This cost includes the system overhead for each region. NITC will work with the agency administrator to determine if the agency's CICS workload or other factors require the establishment of a separate CICS region for that agency. All costs associated with a single-agency CICS region will be charged to that agency.

All CICS resources must be identified to the NITC. Resources are defined to CICS by various tables. When a customer has obtained authorization the following information must be provided:

1. Name and telephone number of customer application project manager.

2. Name of user test load library containing CICS customer application modules.

3. A list of transactions and associated programs that are to be called for each transaction.

4. A list of every program/map in the CICS application system. For programs the programming language must be included.

5. A list of every file in the customer CICS application system including DDNAME, access method, service requests that may be processed against the data set, initial file status, and record format.

6. A list of every DDNAME references by the customer application programs including DSN and DISPOSITION.

7. A list of any other resources required by the application.

CICS Regions

CICS has the flexibility to run multiple regions concurrently. NITC has implemented general-purpose test and production regions for center customers. Also, CICS regions attached to Oracle, DB2, and Inquire/Text are available for customer use.

CICS Telecommunications and Access

The NITC supports the Virtual Telecommunications Access Method (VTAM) and CICS has been implemented for VTAM exclusively.

CICS Security

NITC has implemented the Access Control Facility 2/ Customer Information Control System (CA-ACF2/CICS) security interface, which provides sign on security, and transaction, program and file authorization in an online environment.

CICS Cobol Compile and Map Assemblies

NITC provides procs for the translation, compilation/assembly and linkage of CICS programs and maps.

Cataloged Procedures

Procedure	Description
CICSCOBC	IBM Cobol Compile
CICSMAP	Map Assembly
CICSCEMT	CEMT Batch Facility Procedure

CICS Design Considerations

The design of an online system or application is a complex procedure, involving many

critical elements. Following are some design considerations that may aid in developing a successful CICS online system or application:

System Services:

1. Data access methods, capabilities, characteristics, and services.
2. CICS file control and data management (exclusive control, sharing, data integrity, strings, buffer management, temporary storage, transient data, queuing, etc.).

Data Communications Design:

1. Telecommunications access method, capabilities, and services.
2. Anticipated number of concurrently active terminals.
3. Average size and number of data transactions.
4. Conversational versus Pseudo-conversational transactions and techniques.
5. Mapping services (BMS).
6. Inter-communication Design.

Recovery and Restart:

1. CICS program check detection and recovery.
2. Transaction abend handling.
3. Dynamic Transaction Back out (DTB).
4. CICS Journaling.

Database Design:

1. Hierarchical versus Relational suitability.
2. File utilization and anticipated growth.
3. Batch interface requirements.

CICS Supporting Products

Dynamic Allocation/Deallocation System (DADS)

This online menu-driven facility runs as a CICS transaction and allows dynamic allocation/deallocation of files while CICS is running.

This facility will be available to all customers in the test and acceptance regions, but only to the customer designated project manager of an application system in the production region. This restriction ensures the integrity of the CICS production region. Files may be allocated and deallocated individually or by class.

The Center will build the DADS control file for all regions. Customers must provide DDNAME and DSN for each file.

CEMT Batch Facility

This facility is also available to all customers but is restricted to use with test and acceptance regions for performing the program/map "NEWCOPY" function.

In order to ensure system integrity, customers are not allowed to use this facility in production regions.

A NITC procedure CICSCEMT is available to execute this facility in the batch mode.

Example:

```
//JOBNAME JOB (Accounting,Info),'programmer name',TIME=2,
// CLASS=C,PRTY=2,MSGCLASS=A
//CMTBATCH  EXEC CICSCEMT
//SYSIPT    DD   *
CICS n
CEMT SET PROGRAM(prename) NEWCOPY
/*
//
```

n = number assigned by NITC

NOTE: The SYSIPT DD input data stream must be unnumbered.

References

Online documentation for CICS is available using IBM's BookManager. For help accessing or using BookManager, see the BookManager chapter of this handbook.

Online training is also available for CICS. See the Computer Based Training chapter of this Handbook for more information on available online training.

Advantage CA-DADS Plus User Guide may be purchased from:

Computer Associates International, Inc.
One Computer Associates Plaza
Islandia, NY 11749, USA

CICS/CEMT From Batch manual may be purchased from:

MacKinney Systems
2740 S. Glennstone
Suite 103
Springfield, MO 65804

Chapter 4. JES-Master

Introduction

JES-Master is a spool data management software package for the JES2 environment. JES-Master provides the general customer access to jobs and Sysout data cleared through host-resident security.

JES-Master/SPF runs under IBM's TSO/E and is invoked from the ISPF/PDF Primary Option Menu **=D.J**. ISPF panels have been created to provide an interactive interface to the features of JES-Master.

Features

Customers can access jobs or data sets on the JES2 Spool that they the have security permissions to access. Jobs or individual data sets of the job may be held, requeued, released, cancelled, deleted or submitted. In addition, SYSOUT data set attributes may be modified and the data sets edited, browsed, printed, copied or unloaded to disk.

JES-Master uses ISPF Edit and Browse facilities. Customers are not required to learn a new set of commands since all standard features of ISPF may be used.

In the JES environment, customers can see the status of jobs at a glance. Job list summaries include information on job return codes, size of Sysout data sets, and print queue characteristics for each data set. These features make JES an important monitoring tool for production jobs.

JES allows customers to spot failures within an application job stream soon after they occur, allowing the customer to take prompt corrective action. JES also allows greater flexibility in managing job and Sysout data set characteristics. Jobs and data sets may be requeued under a different class or to alternate destinations.

Customers may perform several different types of selections. The customer may choose to browse a job in its entirety, browse one of the job's Sysout data sets or browse several of the job's Sysout data sets. The selected data sets are viewed as one group.

This handbook chapter provides general information to assist you in accessing and using JES-Master. For more complete information, online tutorials are available by pressing **PF1** after invoking JES-Master.

Establishing Defaults

Upon initial entry to JES-Master, customers should set up default values and selection criteria. These defaults are in effect for all future JES-Master sessions. You may change your defaults at any time. To set up or change your defaults, select **S** from the JES-Master Selection panel.

The main default you will be concerned with will be the JOB list. To change this default, select **1** JOB list from the Default Selection Criteria panel. Set your USER ID to your user ID. Set the Jobname to your user ID*. For instance, if your user ID is USER1, Jobname would be set to USER1*.

You may also want to change your requeue class and destination. To do this, select **0** from the JES-Master Selection panel. You may want to set the Requeue Class to "A" to automatically print. Set the Request Dest parameter to a printer id (i.e. R2900) or LOCAL.

Commands

Primary commands are entered on the command line at the top of the Job list screen (option 1). Some of the most commonly used commands are:

L	**Locate Jobname (Format: L jobname)**
M DATA/LIST	**Set display mode**
NC	**Set or reset the requeue class**
ND	**Set or reset the requeue destination**
•	**Set job selection by source location**
Q	**Set job selection by queue type**
REF	**Reshow or refresh the display**
RF	**Locate a test string in the job list**
S	**Select specific jobname (Format: S jobname)**
SX	**Jobname/jobid select this job(s) for extended display**
X	**Remove deleted jobs from display.**

Line commands are entered in the SEL column beside the desired job. The following lists some commonly used line commands:

A	**Release a held job**
B	**Browse all of a job's Sysout data sets**
BX	**Extended browse**
C	**Cancel a job**
CB	**Display the job's JES2 control blocks**
CD	**Cancel the job with a dump**
CP	**Cancel the job and purge it**
D	**Delete a job's held output**
E	**Edit a job's Sysout**
EX	**Extended edit of all the jobs data sets**

H	**Hold a job**
J	**Display a job's ready output**
L	**Display job information**
M	**Modify Sysout characteristics of all the job's data sets**
P	**Print all of the job's Sysout data sets**
R	**Requeue a job's output (must provide destination ID and/or MSGCLASS=A or system will use requeue defaults of Option O)**
S	**Select a job for data set display**
SX	**Select job for extended display**
SUM	**Display job summary**
T	**Set or modify a job's execution class and/or priority**
U	**Unload job's Sysout**
UX	**Extended Unload**
X	**Exclude job from the display**
SUB	**Edit job for resubmission**

Security

The various facilities of JES-Master are available to individual customers on a selective basis. Each facility is allowed or denied access to any customer through a security exit used in conjunction with NITC existing security.

Chapter 5. Thruput Manager/Job Binding Services

Introduction

Situations that provide services to batch jobs create a number of operational and scheduling problems that are difficult to resolve in a standard MVS/JES2 environment. Thruput Manager/Job Binding Services (TM/JBS) is intended to address the problem of relating jobs that provide resources to each other. TM/JBS manages this type of batch workload.

TM/JBS can be used by software products like CA-IDMS/DC, CICS, ORACLE, DB2, and FOCUS SINK machines. Applications using these products require they be executed on a particular processor and, subsequently, execution of dependent batch jobs must be run on the same processor.

Terms

Activate - A term that refers to the request to make a Binding Agent active. This can be done with operator commands, a JECL statement, or the DD SUBSYS interface.

Agent - A logical element that provides the basic mechanism for scheduling of jobs with Binding requirements.

Binding Agent - A predefined agent that is used by JBS to relate jobs to other jobs or some logical resource.

Bind - Jobs can request to be "bound" to a particular Agent using the BIND mechanism. This is done either with a JECL statement or with JAL.

Control File - A required file that contains the information required for the functioning of Binding Services.

Control File Manager - The component, in JBS, that is responsible for the coordination of all activity to the Control File. It also manages the interface between JES2, JAL and the applications that use the Control File.

DD SUBSYS - The DD Subsystem interface is a standard facility of MVS. Because started tasks cannot use JECL statements, the DD statement interface provides the ability to Activate Deactivate or Bind Agents. Each DD SUBSYS request is the equivalent of a JECL statement for a job.

Job Action Language (JAL) - A simple but powerful control language that allows an installation to describe the actions to be taken with jobs.

Jobs Binding Services (JBS) - JBS addresses the problem of relating jobs that provide resources to each other, such as database managers (e.g. ORACLE and DB2) that provide services to batch jobs with their associated problems. JBS provides a comprehensive set of facilities for the complete management of all aspects of Job Binding.

Job Descriptors - The elementary items that are collected by ThruPut Manager during its job analysis phase. Together they form a comprehensive profile of a job. The Job Descriptors are used in JAL to identify jobs and make decisions about the job.

Job Entry Control Language (JECL) - JES2 control statements, such as /*JOBPARM are known as JECL. JBS introduces some new JECL statements. The JECL control statements provided are unique to TM/JBS; therefore, they are not recognized in a system without ThruPut Manager and will not result in errors that terminate the offending job.

Control Statements

TM/JBS JECL control statements may be placed within the main JCL stream at any location following the JOB statement information. Their parsing, error messages and other system actions are all handled by the JBS of ThruPut Manager.

There can be up to six (6) ACTIVATE JECL statements and up to six BIND JECL statements in a job. The statements allow the activation or binding of Agents either at job initiation or at the initiation of a named step. Customers wishing to bind to multiple agents should contact the NITC Customer Support Center 816-926-6681 for a review of existing system and user agents.

All Agent references are verified at Job Analysis time. The Agents must be already defined; otherwise, the job is failed with a message. Therefore, customers must know valid Agent names to represent resources.

Job Bind Statements (JECL)

//*+JBS ACTIVATE	Used to activate a RELATED Binding Agent. It allows for activation either at job or step initiation. This type of Agent is automatically deactivated at the end of the job.
//*+JBS BIND	This Job Binding statement is used to associate a job with the named Binding Agent. The Agent controls when and where the job is selected for execution.

//*+JBS BIND (COMPOUND)	This Job Binding statement allows you to code several logical BIND statements in one physical BIND statement.

Statement Examples

Activates a Job-Related Binding Agent:

//*+JBS ACTIVATE user-agent

Activate a Job-Related Binding Agent only during a step in the job called STEP3:

//*+JBS ACTIVATE user.agent,FROM=STEP3,TO=STEP3

Bind to a Job-Related Binding Agent:

//*+JBS BIND user-agent

Bind to multiple Job-Related Binding Agents (AND):

//*+JBS BIND user-agent1
//*+JBS BIND user-agent2

or you could use this syntax:

//*+JBS BIND (user-agent1),(user-agent2)

Bind to either Job-Related Binding Agent (OR):

//*+JBS BIND user-agent1,user-agent2

JCL Examples

The following examples demonstrate the mechanism of activating and binding to agents. The application binds to a "system agent" which directs it to execute on a specific processor. During start-up the application activates a "user agent". All batch jobs are dependent on the application bind to the "user agent", thus insuring execution on the same processor.

In this example, job KUSTSTDC (CA-IDMS/CV job) is submitted. This job binds to

system agent SIDMS83 and activates user agent IDMS83.

```
//KUSTSTDC JOB (acct code),.....
//   CLASS=U,PRTY=11,.....
//*
//* The next two lines are JBS Control Statements
//*+JBS BIND SIDMS83
//*+JBS ACTIVATE IDMS83
//*
//DCSYS83 EXEC PGM=IDMS83,.....
```

In this example, job KUSTSTLD, a batch CA-IDMS/DC load job is submitted. It binds to active user agent IDMS83 and executes on the same processor as KUSTSTDC.

```
//KUSTSTLD JOB (acct code),.....
//  CLASS=Q,PRTY=11,.....
//*
//* The next statement is a JBS Control Statement
//*+JBS BIND IDMS83
//*
//JOBLIB  DD  DSN=.....
//LOAD   EXEC PGM=.....
```

User agent IDMS83 must be active before the batch CA-IDMS/DC job can be selected for execution. If user agent IDMS83 is not active the batch CA-IDMS/DC job will be placed in JBS HOLD to await activation of the user agent. In the above example, the job would abort with the an error message, DTM6331A 'BIND' JECL/JAL STATEMENT, AGENT NAME IDMS83 UNDEFINED, if the NITC has not previously defined agent IDMS83.

Chapter 6. Printer Support System (VPS)

Introduction

The VTAM Printer Support System (VPS) is a logical extension of JES-2, filling a void in JES-2 output delivery facilities. VPS routes JES-2 spooled output to any VTAM printer device.

VPS Capabilities

TSO/E - A complete functional replacement for DSPRINT, transparent to the user, is provided via the VPSPRINT TSO/E command processor.

JES - VPS uses a standard JES-2 interface, Process SYSOUT (PSO), for retrieving SYSOUT from the JES-2 spool volumes. This interface provides an independent high performance means of accessing SYSOUT. As SYSOUT becomes available, VPS retrieves it from JES and uses standard VTAM facilities to print it on the appropriate 3270-type printer. The SYSOUT data set is then purged from the JES-2 spool.

SAS/Graph - VPS will print SAS/Graph output, eliminating the need to run SASWTR.

GDDM - VPS will also print graphs created by GDDM version 2 or later, eliminating the need to run ADMPRINT.

Using VPS

There are a variety of methods to accomplish VPS printing:

1. Type **TSO VPSPRINT** at any ISPF/PDF panel command line along with any of the valid parameters for NITC Local print class. Type **TSO PRTVPS** to print to the WSC Local print class.

2. A full screen version of VPSPRINT is available from ISPF Primary Options panel option **N.VP.** Help is available from any ISPF/PDF Command line by issuing the commands **TSO HELP VPSPRINT, TSO HELP VP** or **TSO HELP PRTVPS.**

3. From the ISPF Primary Options panel, type **N.VP** then **1** (VPS Monitor and Control Facility) or **2** (VPSPRINT).

4. Type **TSO VPS** command from the command line of any ISPF panel.

5. Type **VPS** from ISPF option 6 or the READY prompt.

6. Enter the VTAM applid **VMCF**.

7. If it is an application provided on your Televiewing Menu, select **VMCF**. If you do not have this option and would like to request it, forward the request to your agency security officer to have it added to your Televiewing profile.

   VMCF requires an ACF2 logon id and password. A VMCF Tutorial is available by selecting option **T** from the VMCF Primary Option panel.

Chapter 7. Remote Job Entry (RJE) Service

Introduction

This chapter provides an overview of the NITC RJE service. It also provides specific instructions and information needed to connect with and use the NITC RJE service. Detailed command syntax requirements and examples are provided.

General Concepts and Facilities

The RJE customer is similar to a batch customer but is directly connected via telecommunication lines to the computer and can obtain both rapid service and direct communication with other terminals. When JES2 commands are entered, the computer responds with the appropriate action and output. After jobs are entered, they are queued for execution. As jobs finish execution, the output is queued back to the terminal as output unless JES2 commands or statements have specified otherwise.

Parameters with each job govern whether output is to be returned immediately to the terminal, spooled and held until requested, or printed centrally. The job itself contains all necessary JCL for its' processing. The entire job may come through the terminal or customer libraries may be used to supply most of the information.

The JES2 terminal commands and JCL statements can be used to control the input, output and processing of a program. They are also used to send messages and provide information.

To resolve any problems encountered when attempting to utilize RJE services, contact System Network Control Center (SNCC) at (816) 926-6660.

References

This chapter provides the most frequently used JES2/RJE instructions. Customers should also consult the applicable terminal operations manual for instructions as to the operating characteristics and applicable sequences. Technical manuals may be viewed in BookManager. For more information on using BookManager, see the BookManager Handbook chapter.

Terminal Requirements

The typical RJE terminal consists of a card reader and a line printer. Most manufacturers of high-speed terminals have a suitable configuration of their terminal available.

Customers should check with NITC for exact terminal requirements before obtaining a new terminal for use with the NITC RJE service.

In general, a terminal may be connected if it will operate, either by hardware or software, according to IBM conventions as:

- Binary Synchronous
- Continuous Retry
- Extended Binary Coded Decimal Interchange Code (EBCDIC)
- Three (3) Second Initial Time-out
- Transparency Mode

The terminal must operate the same as one of the terminals supported by JES2.

Sign-on Procedure

To use the NITC RJE service, customers must first establish a data communications link with NITC and be identified. Telephone numbers for the available lines can be obtained from the SNCC at (816) 926-6660.

The following are possible receive responses:

1. A high-pitched tone indicates that the system is available and the customer may proceed.

2. A fast busy signal indicates the telephone company cannot connect the call due to insufficient trunk line equipment. Try the number again since this is a telephone company problem and should clear up as lines to the switching center become available.

3. A slow busy signal indicates all lines into the system are busy. Try the number again. If this condition persists, contact the SNCC for assistance.

4. A ringing signal and the telephone will not answer indicate the system is not available. Customers should verify they are dialing the correct number and try again. This could also indicate a modem or port problem. Contact the SNCC for assistance.

When the high-pitched tone is heard, the customer should switch from voice to data mode.

Once the connection is established, the terminal will automatically begin transmitting. The first transmission must be a sign on statement. The format is:

Columns
1 16 73

/*SIGNON RMTnnn PASSWORD

In the above statement, nnn the number assigned by NITC to validate the station to JES2. For single digits, left justify with no leading zeroes (RMT8 instead of RMT08).

The sign on statement should be entered by itself. If the statement is not correct, or if the password is invalid, the sign on will be rejected and the line disconnected. There will be no rejection message to alert you of this condition. If the problem persists, report the problem to the SNCC.

A remote-id password may be obtained by those responsible for terminal operations by your agency security officer. Questions pertaining your remote-id password should be directed to your agency security officer.

Once RJE acknowledges a sign on, the terminal can be used to enter jobs, commands and receive output. If the terminal is inactive for five minutes NITC it will be automatically disconnected to free up the line for other customers. This does not apply to terminals using leased lines.

Command Control Card

The command card is a variable-field control card used to enter JES2 operator commands into the system through the remote console device (which may be a card reader). The format of the card is:

Columns 1-3 /*$
4-71 operator command verb and operands
72 if "N" the command will not be repeated on the operator's console

All command cards must be placed in the input stream prior to any job card. Command cards within jobs will be ignored.

Job Statement Standards

RJE customers must comply with NITC job statement standards when submitting their jobs. Job statements must be coded as follows:

```
//AAABBBBB JOB (CCCCCCCCCCCC,DDDD,E),'FFFFFFFFFFFF',CLASS=E,
//   MSGCLASS=A,NOTIFY=GGGGGG,PRTY=HH,TIME=
```

Entering Jobs

Since operating sequences may vary between terminals, customers should follow those set forth in terminal manufacturers instruction manual. After the last card in a job stream has been read, verification of job entry will be given on the printer.

For example: $13.21.30 JOB 630 JOBNAME NAME FIELD

13.21.30 - time mark when the job entered the JES2 queue.
630 - internal job number assigned by JES2. Use this number to cancel, release or hold the job, or display information about the job.

JOBNAME - jobname field from customer job card.
NAME FIELD - name field on customer job card.

Output Retrieval

The RJE customer has considerable control over the job's output. All job output will normally be returned to the terminal (remote number) the job was submitted from. If the operator desires other distribution, the route or output statement may be used.

Normal Output

Printouts for jobs submitted through a RJE terminal are automatically routed back to the printer facility for the same terminal. If the terminal is shut off from the time a job was submitted and the printout is routed back to the terminal, the printout will stay in a print queue or waiting line. The next time the terminal signed on, the printout will automatically begin printing.

The first and last print pages of a job have a special format and are called separator pages. These pages are used to separate SYSOUT jobs one from another.

Special Forms

When a RJE job has finished running, SYSOUT=A will be handled as normal output. However, SYSOUT=C allows customers to print output on their special forms and must be retrieved as follows:

1. Submit the following command cards:
2. **/*$D F**
   /*$P RN.PR1 /*$T RN.PR1,F=QQQQ (QQQQ is the form number)
   The first instruction stops the computer from presenting any print activity to the RJE terminal. The second card commands the JES2 print scheduler to release the specified form.
3. Change the printer forms.
4. Submit the following command card to start printing:
   /*$S RN.PR1
5. When the special forms have finished printing, submit the following card to hold further prints:
   /*$P RN.PR1
6. Change the printer back to standard forms.
7. Reset the printer.
8. Start the printer.

Transmission Problems

Occasionally the terminal may be disconnected due to line or modem problems. If this occurs repeatedly, please contact the SNCC for assistance with diagnosing the problem. If the line is dropped while output is being transmitted, the JES2 backspace command can be used to reposition the SYSOUT after the terminal is signed on again.

Signing Off

To disconnect from the computer, enter the signoff statement:

/*SIGNOFF

If you have issued a SIGNOFF, you must repeat the sign on process if you wish to enter jobs or receive output again.

Output Routing to Remote Terminals

Customers who desire to have output routed to a remote terminal (e.g. NFC) should do the following:

2. Request NITC to enter a JES2 parameter in JESAPARM to reflect:

   DESTID NAME=xxxx,DEST=Ryyy (xxxx is Form name (e.g., NFC))
   (yyy is Remote number (e.g., R6)

3. When NITC notifies the customer that the above has been accomplished (IPL is required), the following JCL DD statement for a given output will accomplish the output routing:

   //PRINT DD SYSOUT=(A,,xxxx),DCB=( ), (xxxx is Form name (e.g., NFC))
   // DEST=yyyy (yyyy is Remote number (e.g.,R6))

Notify the operator of the remote site to display forms to find the special form requested, and then set the remote printer accordingly.

Chapter 8. LINUX Servers

Introduction

NITC offers customers enterprise class Linux (UNIX) servers with the availability, scalability, and Quality-of-Service our traditional mainframe customers enjoy. Recent offerings by IBM provide the capability to run Linux open source operating systems on our processor complexes.

The advantages of this environment are numerous -

- The ability to economically transport Linux servers to our contracted hotsite provider without purchasing redundant hardware
- Ability to run tens to hundreds of Linux images on a single S/390 platform
- Cost-efficient use of system resources in support of a very large number of Linux servers on a single processor complex
- S/390 reliability, availability, and serviceability now available for the Linux world
- Simplified installation and administration for a multi-Linux S/390 server environment
- Dynamic real-time creation of additional Linux images as needed
- Integrated high-speed connections among Linux images
- Significant reduction in software licensing due to Linux packaging and pricing
- Workloads can grow – taking advantage of the spare capacity of the machine – and shrink again without having wasted the costs associated with an upgrade
- Significant floor space and energy savings can be realized by consolidating servers onto a single S/390 platform
- An optional Linux workspace that is independent of the traditional S/390 workload

One S/390 can do the job of a host of servers scattered across the enterprise, offering greater security and reliability in addition to reducing the complexity of maintaining a large number of servers. S/390's flexibility and management characteristics make it possible to add new Linux servers in minutes rather than days. All of the great flexibility and openness of Linux combined with the qualities of service of S/390 results in an industrial strength Linux environment.

Integrated Facility for LINUX (IFL)

An IFL is a hardware feature available on our CMOS processor complex that allows processor capacity to be dedicated exclusively for Linux use. This capacity is isolated

from the traditional S/390 (z/OS) workload and the addition of capacity and/or software in one environment does not affect the other.

z/VM Operating System (Hypervisor)

z/VM is a tried and true operating system offering a complete server environment for multiple Linux servers on one S/390 processor. z/VM, an easy-to-use, high-performance supervisor, operates within a logical partition and provides the capability to create tens to hundreds of Linux images on a single S/390 server.

The z/VM environment is ideally suited for those who want to move Linux and/or UNIX workloads deployed on multiple servers onto a single S/390 server, while maintaining the same number of distinct server images. This provides centralized management and operation of the multiple image environment, while reducing complexity, easing administration, and lowering costs.

Linux for s/390

Ports of Linux for S/390 are now available from several vendors. NITC supports distributions from Red Hat, SuSe, and TurboLinux. NITC subscribes to support services from IBM Global Services for 7x24 call-in technical support for Linux all of our Linux distributions.

Cost Savings

The Open source environment offers the opportunity to take advantage of "Publicly Licensed" software. Over 4000 packages exist covering the gamut of software features and functionality. NITC can demonstrate multi-tiered applications utilizing packages such as the Apache web server, accessing a MySQL database, with dynamic content produced using the php scripting language; all freely available (with compliance to the Public License restrictions) for use in this (or any) Linux environment.

If you require a more robust environment we can also demonstrate a similar environment utilizing IBM's WebSphere Application Server accessing a DB2 (UDB) or Oracle database as the back end, with dynamic content controlled by JAVA servlets. These traditionally purchased/licensed products will typically result in cost savings for servers in this environment due to the fact that many Linux servers are sharing a single license. Adding more servers does not necessarily require additional software licenses.

Significant hardware savings can be realized when an application has been designed to run on more than one server – to satisfy redundant or multi-tiered designs. No hardware purchases are required to create one or more servers in this environment, and the reliability characteristics of the S/390 platform virtually eliminate the need to purchase redundant hardware – Mean time between failure on the S/390 platform is measured in decades.

Scalability

When a typical UNIX/Mid-range server has reached capacity, the solution is to add another server to the configuration, or to completely replace the existing server with one having greater capacity. The S/390 based Linux server can take advantage of the spare capacity available on the machine, or optionally have additional IFL processors dedicated to the Linux workload and/or, if necessary, have additional processors added to the complex.

Reliability

The S/390 mainframe offers advanced hardware diagnostics. If there is a hardware failure, it is detected and access is routed to another hardware device. If configured, it notifies the IBM support group for a replacement. This reliability is critical for Internet based applications.

Improved Throughput

The S/390 mainframe provides for advanced I/O capabilities for heavy block data transfers. The channel architecture offloads I/O operations onto discrete channel controllers that execute programs written into shared memory from the CPU. This means that file transfer, network, database, and other application performance on a S/390 platform offers a superior architecture making it an ideal system in I/O-intensive climates.

Server Consolidation

Existing customers supporting multiple servers can consolidate those images onto a single S/390 image and realize potential savings in the following areas:

- Better Utilization of Hardware Resources
- Improved Communication Between Servers

The possibility exists for increased communications between a group of servers when migrated to S/390. The communication features built into the z/VM operating system allow for a Virtual “Guest LAN”. This internal "TCP/IP network in a box" eliminates network latency and maximizes bandwidth linkage between Linux and z/OS images, z/OS and z/OS images, or Linux and Linux images within a single CMOS server to support the high-performance data and transaction requirements of e-business.

Server Setup

Since all Linux guests run under the z/VM hypervisor, the S/390 administrator has very little work to do when creating a new Linux image. Linux isn't re-installed for each new account, but copied from a common image. Configuration files are customized to set things like the hostname, IP address, and root password, and the image is ready to use. This is one of the most significant cost-saving features when using Linux on S/390 in a web hosting or Application Server Provider (ASP) application.

Systems Administration

Several servers can be configured to share one common image of the Linux operating system. This reduces the time and complexity of maintaining an accurate inventory of the installed components, and significantly reduces the time necessary to apply security patches and updates – since the administrator only has to apply them once.

Security

The z/VM hypervisor guarantees that independent servers maintain their integrity. Z/VM can also be configured to control the ability of the Linux operating system to write to certain portions of the file system, thereby eliminating the risk of critical components of the operating system being altered.

The S/390 systems come with dedicated cryptography hardware. This enables the server to perform a large number of Secure Sockets Layer (SSL) session handshakes per second. The hardware performs the encryption, and there is no longer a hit in performance when a customer wants to secure valuable data transactions.

Porting to Linux

New and current NITC customers who are looking to move their applications to a more reliable, scaleable platform such as S/390 will have an easier time porting to Linux than to Unix System Services (USS) because this is a true Linux port, it is not emulated or simulated in any way. Source code that is compiled and executed on any other Linux server can be compiled and executed on the S/390 server. In addition, an ever-increasing number of ISV products are available for the Linux environment making the transition from non-Linux platforms easier.

Costs

Virtual Linux servers are priced at Gold Level support and include all of the resources necessary to support a typical Web/Application server. Additional (optional) services may be added to the subscription for hot site testing and recovery, TSM incremental backups, additional (fixed) increments of processing power and/or disk storage.

How To Get A Linux Server

Contact the NITC New Business Manager at 1-888-USE-NITC and ask about a Linux server no-cost trial period.

Chapter 9. WebSphere MQSeries

Introduction

WebSphere MQSeries messaging products enable business applications to integrate and exchange information across many different platforms; sending and receiving data as messages. This product suite manages network interfaces, assuring "once and once only" delivery of messages, manages communication protocols, dynamically distributes workloads across available resources, handles recovery in the event of system problems, and makes application programs portable.

MQSeries provides a consistent multi-platform application programming interface. The latest release (Version 5, Release 3) offers added security using Secure Sockets Layer (SSL), the internet standard for secure communication.

There are also enhanced performance features for applications with system scalability and reliability. This is particularly useful for clustering systems that can share workload. It is now simpler for programmers to use API exits to allow monitoring and implementing local standards.

Features

- Heterogeneous any-to-any connectivity from desktop to mainframe (over 35 platforms supported)
- A comprehensive family of APIs designed to make coding for any messaging task straightforward
- Allows business to integrate disparate islands of automation
- Time independent communication
- Assures one-time delivery
- Supports high volume throughput
- Connect any commercial systems in business today (e Business)
- Ignores network disruptions - important data is always delivered

Visit this link to view Power Point presentation on the features of Websphere MQSeries, prepared by NITC: http://www.ocio.usda.gov/nitc/products/web_sphere.ppt

Policy

MQSeries is available to all NITC customer agencies. An agency desiring to utilize MQSeries should appoint an administrator to coordinate implementation and

management with NITC. This administrator will manage agencies queues and channels within the MQSeries queue managers.

Contact the New Business Manager at 1-888-USE-NITC to inquire about access to Websphere MQSeries.

The cost of the MQSeries queue manager will be prorated based on the CPU usage of each agency.

Security

NITC can provide customers with information regarding security of Websphere MQSeries, including the use of SSL for encryption of data, channel exits for improved connectivity assurances, and an "MQ Security Checklist" for interfacing with the mainframe External Security Manager.

Websphere MQSeries ISPF panels can be configured to allow access to only authorized administrators. NITC uses Computer Associates Access Control Facility (CA-ACF2) security interface to provide security in the mainframe environment. Security rules written for Websphere MQSeries are resource rules written by NITC security staff.

References

MQSeries manuals are available online in BookManager. For help accessing and using BookManager, see the BookManager chapter of the Handbook.

Additional reference materials may be obtained by accessing url: www.ibm.com and entering in the top right SEARCH box: **MQSeries**.

Chapter 1. Managing Data Sets

Introduction

An agency's data is one of its most important resources. In an OS/390 Open System Server environment, data can reside in sequential or partitioned data sets. This chapter provides information needed for effectively managing your data resources.

Sequential data sets are files organized with the data arranged in sequence, from top to bottom or beginning to end. When an application program reads the file, the records are retrieved in the order they are stored.

A partitioned data set (called a PDS or library) consists of a directory that contains information about each member and one or more members containing the data. Each member functions the same as a sequential data set. The members can be processed separately or the entire PDS library can be processed as a single data set. When referring to partitioned data sets, enclose the member name in parentheses immediately following the descriptive qualifier.

Naming Conventions

Each data set is identified by a unique data set name. Data sets must conform to TSO/E and National Information Technology Center (NITC) naming conventions:

- A data set name consists of one or more parts connected by periods. Each part is called a qualifier:
  - **High-level qualifier** (Alias/Index) (Required) – the first qualifier in a data set name is called the high level qualifier and is either your userid or a qualifier provided by your agency.
  - **Customer supplied name** (Required) – this is a name that you create that has meaning to you for processing reference.
  - **Descriptive qualifier**(s) (Required) – a name that you create that has meaning to you for processing reference and differentiates this data set from others in your list. See the table below for descriptive qualifiers.
- Each qualifier must begin with an alphabetic character (A-Z) or the special characters $,#,@.
- The remaining characters in each qualifier can be alphabetic characters, digits (0-9), a hyphen (-), or the special characters $, #, @.

- Each qualifier must be one to eight characters long.
- The maximum length of a complete data set name before specifying a member name is 44 characters, including the periods.

Descriptive Qualifiers

The following is a table of commonly used descriptive qualifiers:

Descriptive Qualifier	Data Set Contains
ASM	ASSEMBLER input
COBOL	COBOL statements
FORT	Fortran statements
FOCUS	FOCUS Data Bases
EXEC	TSO/E commands and REXX instructions
TEXT	Upper and Lower Case Text
DATA	Upper Case Text
CNTL	JCL and SYSIN for SUBMIT command
OBJ	OBJECT module
CLIST	TSO/E commands and CLIST statements
LOAD	LOAD Modules
LOADLIST	Output listing from loader
LIBRN	CA-Librarian
LIST	Output Listings
LINKLIST	Output from Linkage Editor

Default Data Set Names

The system attempts to derive the descriptive qualifier from the characteristics of the command in which the data set name is used. For instance, if no descriptive qualifier is noted for the source data set in a FORTRAN compilation, the system will assign, "FORT," as the descriptive qualifier. If the system has insufficient information to assign a descriptive qualifier, you will be prompted for the information. If the member name of a partitioned data set is not specified, the system will use, "TEMPNAME," as the default.

The following is a list of command names and default descriptive qualifiers associated with each:

COMMAND	INPUT	OUTPUT	LISTING
CALL	LOAD	---	---
COBOL	COBOL	OBJ	LIST
EXEC	CLIST	---	---
FORMAT	TEXT	---	LIST
FORT	FORT	OBJ	LIST
LINK	OBJ	LOAD	LINKLIST
"	LOAD	---	----
LOADGO	OBJ	---	---
"	LOAD	---	---
RUN	FORT	---	---
"	COBOL	---	---
SUBMIT	CNTL	---	---
TEST	OBJ	---	TESTLIST
"	LOAD	---	---

Allocating Data Sets

Storage space must be allocated for a data set at the time that it is created and named, which includes the properties that you assign to the data set. The system sets aside the storage space and catalogs your file name.

Refer to the ISPF tutorial and help features for the most current techniques to allocate data set storage space.

The TSO/E ALLOCATE command may be used to reserve space and define data set properties. Refer to the ISPF tutorial and TSO help features for the most current techniques to applicable to TSO/E commands.

Managing Data Sets

Type **HELP** from the main ISPF panel, for a tutorial on the ISPF panel options given below.

Type **TSO HELP** from ISPF panel option **6** for TSO/E commands information and parameters.

Printing Data Sets - ISPF panel option **=3.6** and TSO/E commands **PRINTDS**, **PRINTOFF** or **DSPRINT.**

Deleting Data sets - ISPF panel option **=3.1** or TSO/E **DELETE** command. To manage your agencies data storage costs, the NITC encourages users to delete unused data sets.

Renaming Data Sets - ISPF panel option **=3.2** or TSO/E **RENAME** command.

Merging Data Sets - ISPF panel option **=3.3** or TSO/E **MERGE** command.

Freeing Data Sets - ISPF panel option **=3.4** or TSO/E **FREE FILE**, **FREE DA(xxxx..nnnn) FI(ssss,tttt)** and **FREE ALL** commands.

Cataloging/Un-cataloging Data Sets – ISPF panel option =3.2.

The system will automatically catalog a data set at the time it is created. Utilities are also available to un-catalog and re-catalog data sets; however, the risk of data set contamination and catalog properties and indexes becoming misaligned should be carefully considered.

The NITC provides a CLIST called, "CATLG," to assist customers in cataloging and un-cataloging data sets. Execute the Clist from ISPF option **6**:

1. Type **%CATLG**
2. Enter your response to the prompts.
3. Type **END** to exit the CLIST at any point.

Copying Data Sets - ISPF panel option **=2 or =3.3** and the TSO/E **COPY** command.

Compressing Partitioned Data Sets - ISPF panel option **=3.1** and **=3.4**.

Backup and Recovery – ISPF panel option **=BU**.

References

Related Sections of this Handbook:

- ABARS
- Data set Backup and Recovery

For more information on any of the functions described in this chapter, see BookManager. For help accessing and using BookManager see the BookManager chapter of this handbook.

In addition, there are also Computer Based Training courses available to assist customers with learning how to use and maintain data sets. See the Computer Based Training handbook chapter for more information on registering and taking on-line training.

Chapter 2. Data On-Line Storage

This chapter is being revised.

For current information, please contact the NITC Storage Management Branch Chief, Larry Reynolds, at 816-926-2316.

Chapter 3. Innovation Access Method (IAM)

This chapter is being revised.

For current information, please contact the NITC Storage Management Branch Chief, Larry Reynolds, at 816-926-2316.

Chapter 4. ICF Catalog

This chapter is being revised.

For current information, please contact the NITC Storage Management Branch Chief, Larry Reynolds, at 816-926-2316.

Chapter 5. Data Tape Storage

This chapter is being revised.

For current information, please contact the NITC Storage Management Branch Chief, Larry Reynolds, at 816-926-2316.

Chapter 6. Data Set Backup and Recovery

This chapter is being revised.

For current information, please contact the NITC Storage Management Branch Chief, Larry Reynolds, at 816-926-2316.

Chapter 1. Job Definition and Requirements

Introduction

JCL statements at NITC serve specific functions. Specifications are listed below.

JOB Statement

Columns
1 12
//**AAABBBBB**
JOB(**CCCCCCCCCCCC**,**DDDD**,**E**),'**FFFFFFFFFFFFFFFFFFF**',
// CLASS=**G**,MSGCLASS=**H**,NOTIFY=**I**,PRTY=**JJ**,
// TIME=(**KKKK**,**LL**),TYPRUN=**MMMMMMMM**

AAA The standard organization code prefix for customer JOB name. Contact your agency security officer to establish a new code.

BBBBB Any five alphanumeric characters chosen by the customers.

CCCCCCCCCCCC The accounting code that is used for billing parameter is required for all JOBs.

DDDD Designation of the distribution of all JOB output at NITC. This parameter is required for all JOBs.

E The JOB status code used by the console operator when the processor time limit has been exceeded. This parameter is NO LONGER required, but if used, it must be a numeric character.

FFFFFFFFFFFFFFFFFFF Programmer/clerk/name or information field. (Customers located in Washington DC or those printing at WSC should use this field for BIN number.)

G The appropriate JOB class specifying resource requirements and turnaround standard. Refer to the Computer Operations chapter of this handbook for valid JOB classes.

H Output class for the JOB. Valid class codes are:

A - Output directed to NITC local printers and/or cardpunch.

R - Output directed to remote batch or interactive terminal
From which the JOB was submitted.
1 - WSC local print

I The TSO LOGON ID to be notified when JOB execution terminates.

JJ Scheduling priority requested for execution. Valid priorities are:

02	- Deferred Processing
11 NORMAL	- 90% to 100% of base charge.
12 INTERMEDIATE	- 133% to 200% of base charge.
13 HIGH	- 233% to 300% of base charge.

KKKK, LL Maximum CPU time required for execution in minutes (KKKK) and seconds (LL) class. See Computer Operations chapter for details on JOB classes.

MMMMMMMM Request for special JES processing. Valid specifications are "HOLD" and "SCAN".

Parameters 'group', 'password', and 'user' in the JOB statement is prohibited.

JOBPARM Statement

Columns
1 3 11
/*JOBPARM COPIES=**AAA**,RESTART=**B**,S=**CCCC**,LINES=**DDDD**

AAA The number of output copies of a job related output that is to be produced (from 1 to 255).
B Specifies whether JOB is to requeued for execution after re-IPL (Y=Yes, N=No)
CCCC CPU authorized to execute the JOB. Valid entries are:

SYSA
SYSB
SYSC

NOTE: Do NOT specify CPU unless instructed by NITC. Initiators may not be assigned to every JOB class on every system.
DDDD Number of lines. Used to override the default

OUTPUT Statement

Columns
1 3 10
/*OUTPUT **AAAA**, COPIES=**BBB**,DEST=**CCCCCC**,FORMS=DDDD

AAAA Form number specifying which output data sets the JES control card applies to.

BBB Number of copies of to be produced (from 1 to 255).

CCCCCC The logical address to which the output is directed.

Remote terminal (Rnnn, RMnnn, or RMTnnn). nnn is the remote terminal number. R0 is equivalent to LOCAL

Local terminal (Unnn). nnn is the local device number.

LOCAL - Any local device

name - a 1 to 8 character (alphanumeric or national) name of a remote or local device (as defined by the system programmer) to receive the output.

DDDD An alphanumeric value indicating the print and punch forms. From 1 to 4 characters.

ROUTE Statement

Columns
1 3 10
/*ROUTE **AAAAA BBBBBB**

AAAAA Type of output -- either "PRINT" or "PUNCH".

BBBBBB The logical address to which the output is directed.

Remote terminal (Rnnn, RMnnn, or RMTnnn). nnn is the remote terminal number. R0 is equivalent to LOCAL

Local terminal (Unnn). nnn is the local device number.

LOCAL - Any local device

name - a 1 to 8 character (alphanumeric or national) name of a remote or local device (as defined by the system programmer) that is to receive the output.

The NITC Security Officer is responsible for assigning and maintaining code prefixes, accounting code numbers, output bins, logonids, and form names. Requests for establishment of new codes should be directed, in writing, to the NITC Security Officer through customer's agency security officer.

Chapter 2. Program Creation and Execution

Introduction

The NITC offers customers facilities for program creation and execution for a wide variety of languages. Programs may be compiled, link-edited and executed in the background (batch) by using command procedures (procs) or in the foreground using TSO/E ISPF panels.

This handbook chapter assumes some familiarity with programming and compilation. If needed, the NITC offers Computer Based Training (CBT) courses that may be helpful. Information on available courses and how to register for them may be obtained by browsing Teleview NEWS option 2 – Computer Based Training. In addition, online documentation is available using IBM's BookManager. For help accessing or using BookManager, see the BookManager chapter of this handbook.

Compiling and Link Editing a Program

The first step in producing a running program is the conversion of a source program (whatever language it is written in) into a form acceptable to the computer. This process is called compilation. The result of compilation is an object module.

Before a program in object module form can be executed, any references to system routines, other routines of the customers, etc., must be resolved. There are two ways to accomplish this: Use of the system's loader, and linkage editing or the LINK command. The linkage editor processes a compiled program (i.e., an object module) and produces an executable load module. The linkage editor can also process more than one object module and/or load module and transform them into a single load module. This is a useful facility when the customer is combining sets of separately compiled routines.

The following will link edit two object modules in PGM1.OBJ and PGM2.OBJ into member PGM3 of LM.LOAD:

```
LINK (PGM1,PGM2) LOAD(LM(PGM3))   COBLIB
                                  FORTLIB
                                  LIB(CUSTOMER'S LIBRARY)
```

The customer may specify a maximum of ten modules to be link edited together. Linkage editor control statements must be used to combine more than ten.

As part of the LINK command, customers must supply the name(s) of the object module(s) to be link edited. When more than one data set is used, they must be enclosed in parentheses. After the names of the object modules, customers should use the load

operand to indicate the member name of a partitioned data set where the load module is to be placed. If the customer-supplied name of the load module data set is omitted, the system assumes it has the same customer-supplied name as the object module (the first module named if the customer provides more than one). If the descriptive qualifier is omitted, the system assumes LOAD. If a member name is omitted, the system assumes TEMPNAME. Customers should not rely on default member names as this can cause frequent losses of data. Customers must supply a library name to be used for resolving external references.

To link edit a load module, the customer must first allocate the library. The command sequence to execute is as follows:

```
ALLOC FI(LIB) DA(OLDLIB.LOAD) SHR
LINK * LOAD(NEWLIB)
INCLUDE LIB(OLDMOD)
NAME NEWMOD(R)
```

To link edit the same object module and place the result in member ONE of a data set MODS.LOAD, the customer enters:

```
LINK PROG2 LOAD(MODS(ONE))
```

NOTE: As in all keywords with parameters, the parenthesis after the keyword LOAD must follow it immediately, with no intervening blanks.

TSO/E and ISPF

ISPF tools are designed to increase productivity in the TSO/E environment for customers of both structured and conventional programming techniques.

The major ISPF functions that facilitate program creation and execution are:

- BROWSE - Display source data or output listings.
- EDIT - Create or change source data in full screen edit mode.
- UTILITIES - Perform ISPF utility functions. (allocate, delete, catalog data sets, etc…)
- FOREGROUND – Interactively compile, assemble, link-edit, or debug.
- BACKGROUND - Compile, assemble, or link-edit. (Automatic Submit to background)
- TSO/E - Enter a TSO/E or CLIST command.

Execute **%ISPF** from the READY prompt to invoke ISPF. From there, select the above function you wish to perform.

NOTE: ISPF is not routinely added to logons - it must be requested.

Executing Programs in TSO/E

Prior to executing a program under TSO/E, you must allocate the data sets required by the program. This associates each program file name with a specific data set.

The RUN command can be used to compile, load and execute the source statements in a data set as follows:

RUN X3.COBOL OR RUN X3 COBOL

The CALL command can be used to execute programs after they have been link edited, or to execute any other program in load form (such as utility). As part of the CALL command, the customer must specify the name of the data set and member that contains the program in load module form. For example, to execute a program contained in member ACCT2 of the partitioned data set MYWORK.FORT type at option 6 of ISPF or the READY prompt:

CALL MYWORK.FORT(ACCT2)

The LOADGO command can be used to execute a program that has not been link edited. The LOADGO command combines the function of the LINK and CALL commands to execute a program without producing a load module. Examples of the LOADGO command are:

LOADGO PROG1 LOADGO (PROG1,COB.OBJ,COB.LOAD(TWO))

Background Processing (Batch)

The NITC provides the following batch procedures (procs) for compiling, link editing and executing programs:

Language	Proc	Description
IBM COBOL	IGYWC	Compile
	IGYWCL	Compile and Link Edit
	IGYWCG	Compile, Load and Execute
	IGYWCLG	Compile, Link Edit and Execute
	IGYWPL	Prelink and Link Edit
	IGYWCPL	Compile, Prelink and Link Edit
	IGYWCPLG	Compile, Prelink, Link Edit and Execute
AD/Cycle C/370	EDCC	Compile
	EDCCL	Compile and Link Edit
	EDCCLG	Compile, Link Edit, and Execute
	EDCCLIB	Compile and add the Object Module to a Library

	EDCCPL	Compile, Prelink, and Link Edit
	EDCCPLG	Compile, Prelink, Link Edit and Execute
PL/I for MVS and VM	IEL1C	Compile
	IEL1CG	Compile and Execute
	IEL1CL	Compile and Link Edit
	IEL1CLG	Compile, Link Edit, and Execute
VS Fortran	VSF2C	Compile
	VSF2CL	Compile and Link
	VSF2CLG	Compile, Link and Execute
	VSF2LG	Link and Go
	VSF2G	Execute only
	VSF2CG	Compile and Load
	VSF2L	Load only
REXX	RXTCL	Compile and Link

For more detailed information on specific programming languages, see Compilers and Related Products section of this handbook.

Sample JCL

C/370 Compile

```
//JOBNAME  JOB (ACCOUNT INFO),'JOB INFO',
 //        CLASS=D,MSGCLASS=R,MSGLEVEL=(1,1),
 //        NOTIFY=XXXXXX,PRTY=11,TIME=5
 //STEP1 EXEC EDCC,CPARM='SOURCE'
 //COMPILE.SYSIN DD DSN=YOUR.SOURCE.LIB(MEMBER),DISP=SHR
 //COMPILE.SYSLIN DD DSN=YOUR.OBJ.LIB(MEMBER),DISP=SHR,UNIT=
```

C/370 Link

```
//JOBNAME  JOB (ACCOUNT INFO),'JOB INFO',
 //        CLASS=D,MSGCLASS=R,MSGLEVEL=(1,1),
 //        NOTIFY=XXXXXX,PRTY=11,TIME=5
 //STEP1 EXEC EDCL,LPARM='AMODE31,LIST,MAP,XREF'
 //LKED.SYSLIN DD DSN=YOUR.OBJ.LIB(MEMBER),DISP=SHR
 //LKED.SYSLMOD DD DSN=YOUR.LOAD.LIB(MEMBER),DISP=SHR,UNIT=
```

C/370 Compile and Link

```
//JOBNAME  JOB (ACCOUNT INFO),'JOB INFO',
 //        CLASS=D,MSGCLASS=R,MSGLEVEL=(1,1),
 //        NOTIFY=XXXXXX,PRTY=11,TIME=5
 //STEP1 EXEC EDCCL,CPARM='SOURCE',LPARM='AMODE31,LIST,MAP'
 //COMPILE.SYSIN DD DSN=YOUR.SOURCE.LIB(MEMBER),DISP=SHR
 //LKED.SYSLMOD DD DSN=YOUR.LOAD.LIB(MEMBER),DISP=SHR,UNIT=
```

Foreground Processing (Interactive)

Interactive language processing may be accomplished by accessing TSO/E ISPF option 4 (Foreground) and selecting the desired language.

Some of these include: VS Fortran, PL/I, C/370, REXX, and IBM COBOL.

NOTE: Although the ability still exists to compile and execute OS/VS COBOL and COBOL/II programs, customers should be aware that IBM no longer supports these versions of COBOL. NITC will continue to install these compilers and run-time libraries as long as this can be accomplished. However, IBM will not offer compiler upgrades or problem fixes should users of OS/VS or COBOL/II experience compile or execution problems.

Conversion to the current IBM COBOL is strongly recommended. Mhtran-2 is a COBOL translator product that can help customers in conversion. However, many COBOL/II and OS/VS COBOL programs may not require source code changes, and will compile correctly with the current IBM COBOL compiler. See the Handbook chapter on Mhtran-2 for more information or contact the NITC Customer Support Center at (816) 926-6681.

Chapter 3. CA-JCLCheck

Introduction

CA-JCLCheck is a Computer Associate provided software package used to validate MVS JCL before it is submitted for execution.

The major features of CA-JCLCheck are:

- Complete syntax checks on the JCL coding. The input libraries can be sequential, partitioned data sets (PDS) or Librarian files.
- Execution time errors resulting in system abends caused by JCL problems are flagged (including missing or invalid programs, missing data sets, and incorrect DCB information).
- Other common errors are checked, such as incorrect order of cataloged procedure overrides.
- Potential ACF2 Security Violations are flagged.
- JCLNEAT component used to reformat JCL statements according to user defined specifications.
- Data set allocation requests are checked for efficiency of track utilization based on block size.

NOTE: CA7, and CA11 interfaces are NOT installed.

Invoking CA-JCLCheck

CA-JCLCheck can be invoked:

1. Interactively while editing a data set.
2. From ISPF panel using JCL-Check/SPF.
3. Executing a batch job to the system with the results generated to Sysout.

Interactive Access

To invoke CA-JCLCheck interactively, you must be in edit mode. On the command line, issue the **!JCK** command. CA-JCLCheck will then process the JCL statements and display the results on the screen. An ISPF Program Function (PF) key can also be set up to issue this command.

JCL-Check/SPF can be invoked by typing **=D.JC** from an ISPF panel. From ISPF panels, JCL-Check/SPF allows you to: (1) Invoke JCLCheck in the foreground, (2) Invoke JCLCheck in the foreground and submit the JCL analyzed to background if there

are no errors found, (3) Submit a batch job to invoke JCLCheck, or (4) Interface with Scheduling System.

Specify the data set containing the JCL you wish to analyze. If the data set is a partitioned data set, you may specify a member name or leave the field blank and press ENTER to display the Member Selection Panel. Then, select the members to be processed from that panel. The order for processing the members may also be specified.

You can change all of the JCLCheck run-time options. Your options are saved in your ISPF profile. You may use the menus to change any of the options, or you may override them directly from the main panel.

Both the SYSTERM and SYSPRINT files may be browsed. A panel to invoke printing reports can be requested to generate hardcopy.

For more information, type HELP from the CA-JCLCheck/SPF Menu.

Batch Execution

For a batch job, the following example can be used:

```
//jobname  job  (account,dist),name,CLASS=K,TIME=01,
//   PRTY=11,MSGCLASS=R,MSGLEVEL=(1,1)
//USERLIB  JCLLIB  ORDER=(SYS90R.JCLCK.CAIPROC)
//STEP01  EXEC CAZ1JCHK,
//    OPTION='O(OPTS)'
//SYSPRINT DD  SYSOUT=*
//JCHKFREE DD  DUMMY
//JENDFREE DD  DUMMY
//SYSGRAPH DD  SYSOUT=*
//SYSIN    DD  DSN=jcl.data set.name,DISP=SHR
//OPTS     DD  *
 ERROR(INLINE)
 COND
 CTLSCAN
 DESTCHK
 PXREF
 SPACE
 AUTOPROC
 RPTRPT
 XREF
 GRAPH
 SE(12)
/*
```

Available Options

There are many options available for CA-JCLCHECK. Some of these include:

- EXCLUDE – specify members to be excluded from input.
- INPUT - specify alternate ddname used to describe primary input.
- LIBRARIAN – specify input be retrieved from a CA-LIBRARIAN master file.
- INCLUDE - members to be included from input.
- PDS - specify primary input will be a partitioned data set.
- OPTIONS – specify additional options will be input via a card-image file.
- PROC – specify an alternate ddname for the procedure library.
- RDR - specify an alternate ddname used with output job streams generated by SUBMIT option.
- SUBMIT - requests automatic submission of error-free job streams.
- COBEXIT - activates COBOL User Exit facility.
- COND - activates condition code simulation.
- CTLSCAN - requests read and interpretation control statements for IBM utility programs.
- DESTCHK - requests validation of remote destination names on DD statements and JES control cards.
- MVSLEVEL - Use these rules to scan a given set of RUNJCL. Indicates the version of MVS syntax rules.
- PXREF - activates the program and member existence checking and reporting feature.
- RUNTIME - Requests the runtime checking of JCL.
- SPACE - requests DASD space efficiency analysis.
- STANDARD - names the standard to be used for scanning JCL. This is defined by the user through the Job Control Standards feature.
- SYNTAX - requests the syntax checking of JCL.
- AUTOPROC - Requests the cataloged procedure library definitions be extracted from the JES2 address space.
- DEQUE - free selected enqueued data sets.
- FEATURE - special features in effect for this execution.
- INDEX - restricts catalog lookup and data set existence to specified high-level index.
- SYSUID - specifies the User ID value to be used in replacement of the system symbolic parameter &SYSUID.
- PDSRELATE - specifies the input library is to be treated as a series of consecutive, related job streams.
- SXREF - activates related job data set reference checking and reporting.
- SECURITY - activates ACF2 checking.
- VSAM - requests support of VSAM catalog management.
- RUNCPU - specifies a CPUID to use for the security validation process.
- RUNDATE - specifies a future run date and time to use to determine if access will be granted at that time.
- USER - specifies the user ID to use in conjunction with security checking.
- LIST - writes reports to primary report file (SYSPRINT).
- TERM - generates the terminal report file, containing only the erroneous statements and messages (SYSTERM).
- CCLIST - lists the first few cards of every input stream data set.
- ERROR - specifies the location of error messages.
- FULLLIST - generates listing #2.
- LINECNT - specifies the number of lines to be printed on each page.
- MSGLEVEL - permits the suppression of information messages.
- SEV - specifies the minimum severity level of errors.

Reports

Several reports are available, depending on the options used above:

Report #1	Job stream JCL, Option JOB
Report #2	Listing of Merged JCL, Option FULLLIST
Report #3	Data set Cross Reference, Option XREF
Report #4	Program Cross Reference, Option PXREF(RPT)
Report #5	Report Option RPTRPT
Report #6	Error Messages, Option ERROR
Report #7	Summary Data set Cross Reference, Options SXREF and XREF
Report #8	Summary Program Cross Reference, Options SXREF and PXREF
Report #9	Summary Report, Options SXREF and RPTRPT
Report #10	Flow Diagram Report, Option GRAPH
Report #11	Procedure Cross Reference Report, Option PROCXREF

References

JCL-Check manuals may be viewed in BookManager. For information on how to access and use BookManager, see the BookManager chapter of this handbook.

The following reference manuals may also be ordered directly from Computer Associates:

CA-JCLCHECK MVS User Guide
CA-JCLCHECK Systems Programmer Guide
CA-JCLCHECK Examples Guide
CA-JCLCHECK Message Guide
CA-JCLCHECK Master Index

To order the manuals, write or call:

Computer Associates International
2291 Woodoak Drive
Herndon, VA 20171

Attn: Federal Division

(703) 709-4500

Please indicate that JCLCHECK is installed at the National Information Technology Center to allow for direct ordering and billing.

Additional reference information may also be obtained by accessing url: http://www.ca.com/ and entering in the SEARCH box: **jclcheck**.

Chapter 1. IBM OS Utilities

Introduction

IBM supplies many different utility programs that can be used for common data processing functions. For example, a common data processing task is to copy a sequential file. Although a programmer could write a program to accomplish this, it would be a waste of programming effort to create a new program each time you want to copy one sequential file to another. An IBM utility program can accomplish this task much quicker.

Utility Programs

Although there are many utility programs, the following are the most frequently used:

Program Name	Function
ICEGENER	Copy data sets
IEBCOMPR	Compare data sets
IEBUPDTE	Delete/Insert/Modify records in data sets
IEBPTPCH	Edit/print/punch data sets
IEBCOPY	Copy/Merge data sets
IDCAMS	VSAM general-purpose utility program
IEHLIST	Lists system data (catalogs, VTOCs, PDS directories) .
IEHMOVE	Copies data sets
IEHPROGM	Catalogs, uncatalogs, scratches, or renames data sets
IEFBR14	Catalogs, uncatalogs, scratches, or creates null data sets
Sort/Merge	Sorts or merges sequential data sets
IEBISAM	Prints, copies, or unloads ISAM data sets

OS Utility Examples

Copying a sequential file –

```
//jobname    (job statement)
//           EXEC PGM=ICEGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1   DD DSN=data set.copied.from,DISP=OLD
//SYSUT2   DD DSN=data set.copied.to,DISP=OLD
//SYSIN     DD DUMMY
```

Print a member of a partitioned data set -

```
//jobname    (job statement)
//             EXEC PGM=ICEGENER
//SYSPRINT   DD DSN=data set.to.print(member),DISP=SHR
//SYSUT2      DD SYSOUT=A
//SYSIN         DD DUMMY
```

Preallocate and catalog a fixed block data set -

```
//STEPNAME EXEC PGM=IEFBR14
//SYSUT1   DD   DSN=DSNAME,UNIT=STORE,DISP=(NEW,CATLG),
//    DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120),
//    Space=(TRK,(1,1))
/*
```

Catalog an uncataloged data set -

```
//STEPNAME EXEC PGM=IEFBR14
//SYS001   DD   DSN=DSNAME,UNIT=STORE,DISP=(OLD,CATLG),
//         VOL=SER=SERIAL
/*
```

Scratch cataloged data sets -

```
//STEPNAME EXEC PGM=IEFBR14
//DD1     DD   DSN=DSNAME1,DISP=(OLD,DELETE)
//DD2     DD   DSN=DSNAME2,DISP=(OLD,DELETE)
//DD3     DD   DSN=DSNAME3,DISP=(OLD,DELETE)
/*
```

Comparing two sequential data sets –

```
//jobname   (job statement)
//             EXEC PGM=IEBCOMPR
//SYSPRINT    DD SYSOUT=A
//SYSUT1       DD DSN=first.data set,DISP=OLD
//SYSUT2       DD DSN=second.data set,DISP=OLD
//SYSIN          DD DUMMY
```

Copying a VSAM file –

```
//jobname  (job statement)
//             EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//INVTRAN DD DSN=outfile.data set,DISP=OLD
//INDD       DD DSN=infile.data set,DISP=OLD
//SYSIN      DD *
 REPRO INFILE(INDD) –
            OUTFILE(INVTRAN)
/*
```

Listing a VSAM catalog using IDCAMS –

```
//jobname   (job statement)
//             EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
```

```
//SYSIN       DD *
 LISTCAT CLUSTER –
             NAME
/*
```

References

BookManager contains IBM manuals with more information about IBM OS Utilities. See the BookManager chapter of this handbook for information on accessing and using BookManager.

Online training classes are also available on OS Utilities. See Televiewe NEWS for a list of available training courses on this and other topics or call the NITC Customer Support Center at (816) 926-6681. For help accessing and using Televiewe NEWS, see the Televiewe chapter of this handbook.

Chapter 2. TSO/E Superset

Introduction

TSO/E Superset is a group of utilities from Applied Software, Inc. for use on the IBM MVS operating system. These utilities enable users to easily edit and manipulate data sets, programs and job output from a TSO/E terminal.

Accessing TSO/E Superset

TSO/E Superset Utilities may be invoked from the ISPF panel, **=3.AU** or by typing the Superset command at ISPF, option 6 or the READY prompt.

Superset Commands

Superset Commands	Description
COMPARE	Compare two files or subsets of two files. The files may be displayed in vertical or horizontal split-screen format. Subcommands are provided for searching and file positioning.
COPY	Copy sequential data sets, GDG data sets, members of or entire partitioned data sets (PDS).
FORMAT	Provides output document processing according to control statements embedded within the text.
FSHELP	Display TSO Help.
LIST	Displays the contents of sequential data sets, GDG data sets, and PDS's.
LISTDDN	Displays the content of sequential data sets, GDG data sets, and PDS's.

LISTJES	Displays job output from JES held sysout queues.
MERGE	Combines all or a portion of a data set with another sequential data set, GDG file, or PDS. The input data may be inserted as a block, or the sequence fields of both data sets may be used to determine the insertion sequence.
PRINTDS	Prints sequential data sets, GDG files, members of or entire PDS's. Sysout class, copies, forms and other print parameters may be specified. All print parameters are retained in the profile data set.
TSOSORT	Sorts sequential data sets, GDG files, or PDS's. It allocates data sets, generates control statements and invokes the IBM SORT to sort the data.
MERGE Subcommand of EDIT	Merges a sequential data set, GDG file, PDS, or the EDIT work file into the data set being edited. All functions of the MERGE command are available.
FORMAT Subcommand of EDIT	Formats the edit work file data set. All functions of the FORMAT command are available.
TSOSORT Subcommand of EDIT	Sorts the contents of the edit work file. All functions of the TSOSORT command are available. If the edit data set is numbered, TSOSORT changes the data set to NONUM before sorting.

Profile Data Set

TSO Superset Utilities use a profile data set to retain user-dependent parameters, such as, program function key assignments and the subcommand separator character. They are assigned default values assigned that may be changed by the user. When these parameters are changed, the new values are saved in the profile data set. The saved values are recalled whenever the TSO user is active.

Subcommands

The following subcommands are available from **(C)OMPARE, FS(H)ELP, (L)IST, LIST(D)DN, and LIST(J)ES.** The chart below lists a few of the most common subcommands.

The letters in parenthesis above indicate if a subcommand may be executed from a utility program. For instance, CHLDJ indicates that the subcommand may be executed from all of the above utility programs. Subcommands that manipulate sysout files are available only from LISTJES.

Superset Subcommands

Subcommand	Description	Utilities
BOTTOM	Display last page/line of data.	CHLDJ
CANCEL	Terminate processing for the current component.	CHLDJ
CHAR	Display data in character format.	LDJ
COL	Display column template or specified column range.	CLDJ
COLOR	Turn color support ON or OFF	CHLDJ
CRUNCH	Generate compressed output.	LDJ
DELETE	Delete the sysout data set or entire job.	J
DOWN	Move display down.	CHLDJ
END	Terminate processing for the current data set.	CHLDJ
ENDJOB	Terminate LISTJES for the selected job.	J
EXIT	Terminate processing for the current data set.	CHLDJ

FB	**Display previous sysout file.**	**J**
FF	**Display next sysout file.**	**J**
FIND	**Find character string.**	**CHLDJ**
FINDP	**Equivalent to FIND 'string' PREV.**	**CLDJ**
FUNCTION	**Display subcommand function from HELP.**	**H**
GO	**Begin or resume data set comparison.**	**C**
HB	**Half-page backward.**	**CHLDJ**
HELP	**Display the specified subcommand help document.**	**CHLDJ**
HEX	**Display data in hexadecimal format.**	**LDJ**
HF	**Half-page forward.**	**CHLDJ**
HSPLIT	**Split the display in horizontal format.**	**C**
LEFT	**Scroll display to the left.**	**CLDJ**
MD	**Move display to a specified line.**	**CLDJ**
OPERANDS	**Display subcommand operands from HELP.**	**H**
PB	**Page backward.**	**CHLDJ**

PF	**Page forward.**	**CHLDJ**
PFK	**Equivalent to KEYS.**	**CLDJ**
PRINT	**Print page, range or entire data set.**	**HLDJ**
QUIT	**Terminate processing for current command.**	**CHLDJ**
RECALL	**Display previous subcommand.**	**CHLDJ**
RELEASE	**Release some or all unused space in data set.**	**J**
REPEAT	**Re-execute the previous subcommand string "n" times.**	**CHLDJ**
REQUEUE	**Requeue sysout data set or the entire job.**	**J**
RIGHT	**Scroll display to the right.**	**CLDJ**
SAVE	**Save sysout to OS data set.**	**J**
SEARCH	**Re-search the job queue.**	**J**
SEPC	**Change subcommand separator character.**	**CHLDJ**
STSORT	**Sort the job name list in reverse order.**	**J**
SUBCMDS	**Display subcommand names from HELP.**	**H**
SYNTAX	**Display subcommand syntax from HELP.**	**H**
TOP	**Display first page of data.**	**CHLDJ**

TSO	**Execute TSO command or CLIST .**	**CHLDJ**
UP	**Move display up.**	**CHLDJ**
VSPLIT	**Split the display in a vertical format.**	**C**

References

To order a TSO/E Superset User's Guide, contact:

Applied Software, Inc.
3655 Route 202, Suite 115
Doylestown, PA 18901

(215) 348-3500

Chapter 3. TSO/E PRINTOFF

Introduction

TSO/E PRINTOFF and PRTO are TSO/E Utility programs that may be used to obtain printed copies of data sets through foreground copying to sysout.

The PRINTOFF command can direct a print copy of a data set to the Local printer. The PRTO command is used to direct a printed copy of a data set to a WSC Local printer.

Listings of this type may also be directed to remote site VPS printers. For more information on VPS printers, see the Printer Support System (VPS) chapter of this Handbook.

The PRINTOFF and PRTO commands prints data sets which are sequential or partitioned, blocked, or unblocked, with fixed record length less than or equal to 255 (254 is without carriage control) or variable record length less than or equal to 251 (250 if without carriage control).

Additional information on this utility may be obtained in native TSO or from any ISPF panel by typing **TSO HELP PRINTOFF** or **TSO HELP PRTO**.

Syntax

PRTO ('DSLIST') CLASS('CLASS') DEST('STATIONID')
COPIES('NNN') HOLD/NOHOLD
FORM('FORMNAME') FCB('FCBNAME') WTR('WTRNAME')
LIST/NOLIST PRINT/NOPRINT
FOLD/NOFOLD VOLUME('VOLSER')

Required -- 'DSLIST'
Defaults -- CLASS(1), COPIES(1), NOHOLD, LIST, PRINT, NOFOLD

PRINTOFF ('DSLIST') CLASS('CLASS') DEST('STATIONID')
COPIES('NNN') HOLD/NOHOLD
FORM('FORMNAME') FCB('FCBNAME') WTR('WTRNAME')
LIST/NOLIST PRINT/NOPRINT
FOLD/NOFOLD VOLUME('VOLSER')

Required -- 'DSLIST'
Defaults -- CLASS(A), COPIES(1), NOHOLD, LIST, PRINT, NOFOLD

PRINTOFF Operands

Operand	Description
DSLIST	Data set list contains the names of data sets to be printed.
CLASS('CLASS')	Sysout class in which output is to be printed. Default is 1.
COPIES('NNN')	Number of copies to be printed. Default is 1.
DEST('STATIONID')	Remote station to which sysout data sets are to be routed.
HOLD	Output is to be placed in a hold queue upon deallocation.
NOHOLD	Output is not to be placed on a hold queue upon deallocation.
LIST	Member names only are listed unless print is explicitly stated.
NOLIST	Member names are not to be listed (only printed unless NOPRINT is explicitly stated).
PRINT	Members are only to be printed (not listed unless LIST explicitly stated).
NOPRINT	Members are not to be printed (only listed unless NOLIST is explicitly stated).
FOLD	Output is to be converted to upper case prior to printing.
NOFOLD	Output is not to be converted to upper case prior to printing.
VOLUME('VOLSER')	Volume serial of volume on which data sets to be printed are found. This volume serial will be used for all data sets specified in the data set list.

FORM('FORMNAME')	Form Name (1-4 Characters). Default NONE.
FCB('FCBNAME')	FCB Image (1-4 Characters). Default NONE.
WTR('WTRNAME')	Special Writer (1-8 Characters). Default NONE.

Chapter 4. DSLIST Utility

Introduction

DSLIST is a utility invoked using from a TSO/E CLIST or BATCH procedure. It is used to list data set names that are on tape or disk, cataloged or uncataloged. It will also list data set names for multi-file tapes.

The DSLIST output will list data set names, devices (D or T), catalog status (C, N, or W), creation dates, file sequence numbers (0 for disk data sets), volume sequence numbers, and volsers.

This utility will not list data sets that are migrated at level 2 (ML2).

There are several selection parameters that may be used to limit the data set list. You may select:

- Partial data sets - match on the first n characters a data set name.
- Specific volsers - match on a single volser or volser group.
- Device types - specify either disk or tape only.
- Catalog status - specify cataloged only, non-catalog non-duplicates only, or non-cataloged duplicate data sets only.

DSLIST CLIST

Type the following at ISPF option 6 or the READY prompt:

%DSLIST or %DSLIST P('dsn,lev,vol,dev,cat') or %DSLIST P(?)

DSLIST Parameters

Parameter	Description
DSN	Specifies the name of the data set the output list is written to. This data set must be pre-allocated. Default – output goes to sysout.
LEV	Specify a data set name mask for the data sets to be listed. It must be at least three characters in length. If you wish to limit the list of data sets to a specific prefix, then the period must be included. Default – current prefix specified in your TSO/E profile.
VOL	Specifies a volser mask. This can be used to limit the data set list to a specific volume or to a group of volumes. For example, to obtain

	a list of only migrated data sets, you could specify MIGRAT as the volume.
DEV	Options are D or T. D will list disk data sets only. T will list tape data sets only. Default – both disk and tape data sets.
CAT	Options are C, N or W. C lists cataloged data sets only. N lists non-cataloged, unique data sets. W will list non-cataloged, duplicate data sets.

NOTE: When specifying a dsn that is fully qualified, it is necessary to enclose it in parenthesis and quotes.

When pre-allocating the output data set, specify an LRECL of 72. An appropriate block size should be used.

If you specify **%DSLIST P(?)**, you will be queried for each parameter individually.

CLIST Examples

List all data sets; both tape and disk that begin with prefix ABC01:

```
%DSLIST
```

List all tape data sets that begin with ABC01.MA and write the list into existing data set ABC00.DSLISTT.DATA.

```
%DSLIST P(('ABC00.DSLISTT.DATA'),ABC01.MA,T')
```

List all migrated data sets for your agency:

```
%DSLIST P(',ABC,MIGRAT')
```

List all non-cataloged, non-duplicate data sets for prefix ABC00.

```
%DSLIST P(',ABC00.,,,N')
```

DSLIST Batch JCL

```
//job statement …….
//stepname EXEC DSLIST,LEV=lev,VOL=vol,DEV=dev,CAT=cat
//DSLISTSL DD DUMMY   to suppress printing of data set list (optional)
//DSLISTPR DD ....    pointing to user data set   (optional)
```

See the parameter table above for a description of the EXEC statement parameters.

If the DSLISTSL DD card is specified, it must point to a data set with an LRECL of 72. An appropriate block should be used.

Batch JCL Examples

List all data sets; both tape and disk that begin with prefix ABC01. -

```
//job statement…..
//stepname EXEC DSLIST,LEV='ABC01.'
```

Write a list of all tape data sets that begin with ABC01.MA into data set ABC00.DSLISTT.DATA and suppress the listing of the data sets -

```
//job statement……..
//stepname EXEC DSLIST,LEV='ABC01.MA',DEV=T
//DSLISTSL DD DUMMY
//DSLISTPR DD DSN=ABC00.DSLISTT.DATA,DISP=(OLD,KEEP)
```

List all migrated data sets in agency ABC -

```
//job statement….
//stepname EXEC DSLIST,LEV=ABC,VOL=MIGRAT
```

List all non-cataloged, non-duplicate data sets for prefix ABC00. -

```
//job statement….
//stepname EXEC DSLIST,LEV='ABC00.',CAT=N
```

Chapter 5. DFSORT

Introduction

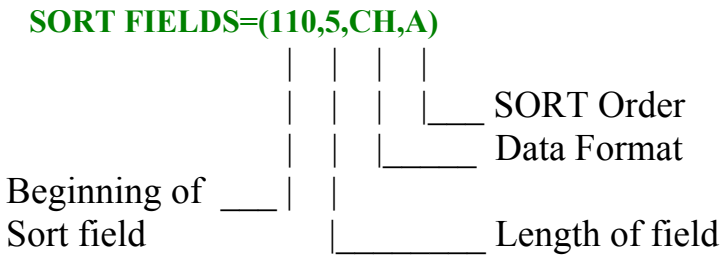
IBM's DFSORT is a licensed program used to sort, merge, and copy information. In addition to these three basic functions, you can also control which records to keep, edit and reformat your records, sum numeric information, and/or create one or more output data sets.

Creating and Running DFSORT Jobs

You can run DFSORT jobs directly with a JCL EXEC statement. Or you can call DFSORT from within a program. To run DFSORT with the JCL EXEC statement, a control statement is used to describe the control fields and the order in which you want them sorted. The control statements you write are part of the input stream read by the SYSIN DD statement in the JCL.

SORT Control Statement

The control statement must be coded between columns 2 and 71 in your JCL. Your control statement should look like the example below:



If you cannot fit your control statement between columns 2 through 71, you can continue it on the next line by ending the line with a comma followed by a blank. DFSORT will then treat the next line as a continuation. The continuation can begin anywhere between columns 2 through 71.

```
SORT FIELDS=(110,10,A,145,17,A,
      1,75,A),FORMAT=CH
```

You can sort data in several different formats and in either ascending or descending order. The most common data formats and the codes you use to specify them are shown below.

Data Format Codes

- EBCDIC (Character) CH
- Binary (Numeric) BI
- Zoned Decimal (Numeric) ZD
- Packed Decimal (Numeric) PD

Sort Order

- Ascending A
- Descending D

You can also specify multiple control fields as shown in the example below. When you specify two or more control fields, you specify them in the order of greater to lesser priority.

```
SORT FIELDS=(110,5,CH,A,115,5,CH,A,145,15,CH,A,160,2,CH,A)
```

When you specify two or more control fields, you specify them in the order of greater to lesser priority.

Sample SORT JCL

Below is some sample JCL that will run DFSORT. It assumes the input and output record lengths are the same.

```
//job statement…
//SORTSTP1 EXEC SORTD6,NCYL=200,SCYL=20,REG=4096K
//SORTIN    DD DSN=YOUR.INPUT DSN,DISP=SHR
//SORTOUT   DD DSN=YOUR.OUTPUT DSN,
//          DISP=(NEW,CATLG,DELETE),
//           DCB=(RECFM=FB,LRECL=30,BLKSIZE=27990),
//           UNIT=LARGE,SPACE=(CYL,(1000,150),RLSE)
//SYSIN     DD *
  SORT FIELDS=(37,8,CH,A)
```

This JCL reads in a file referenced by the DDNAME SORTIN, sorts it based on the parameters in the SORT FIELDS statement and outputs a sorted file referenced by the DDNAME SORTOUT. The input file is sorted by a field starting in position 37 for a length of 8. The “CH” defines the field as character and the “A” indicates that the data will be sorted in ascending order.

MERGE Control Statement

The format of the MERGE statement is the same as that of the SORT Statement as shown in the example below:

```
MERGE FIELDS=(110,5,A,1,75,A),FORMAT=CH
```

The JCL needed for a merge is similar to that of a sort, with the following exceptions:

- The SORTWKnn DD statement is not used.
- The SORTINnn DD statement is used instead of the SORTIN DD statement to define the input data sets. One SORTINnn DD statement is needed for each data set being merged. The value nn in SORTINnn is a number from 01 to 16, indicating the number of data sets to be merged.

Sample MERGE JCL

```
//job statement…..
//SORT    EXEC SORT
//SYSOUT   DD   SYSOUT=*
//SORTIN01 DD   DSN=YOUR.INPUT.FILE1,DISP=OLD
//SORTIN02 DD   DSN=YOUR.INPUT.FILE2,DISP=OLD
//SORTOUT  DD   DSN=YOUR.OUTPUT.FILE,DISP=OLD
//SYSIN    DD   *
   MERGE  FIELDS=(20,8,A,1,5,A),FORMAT=CH
/*
```

COPY Control Statement

The COPY statement allows you to copy data sets directly without performing a sort or merge. The COPY statement can be used with all of the other DFSORT control statements except the SUM statement.

The COPY statement can be specified on the SORT, MERGE, or OPTION statements as shown below:

```
SORT FIELDS=COPY
MERGE FIELDS=COPY
OPTION COPY
```

The JCL for a copy is similar to that of a sort, except that the SORTWKnn DD statement is not used.

Sample COPY JCL

```
//job statement…
//SORT    EXEC SORT
//SYSOUT   DD   SYSOUT=*
//SORTIN   DD   DSN=YOUR.INPUT.FILE,DISP=SHR
```

```
//SORTOUT  DD   DSN=YOUR.OUTPUT.FILE,DISP=OLD
//SYSIN    DD   *
      SORT FIELDS=COPY
/*
```

The MERGE FIELDS=COPY or OPTION COPY can be used in place of the SORT FIELDS to produce the same results.

SUM Control Statement

The SUM statement works in conjunction with the control fields of the SORT statement. The contents of the summary fields are summed only when the contents of the control fields are of the same data type.

```
SORT FIELDS=(10,8,CH,A)
SUM FIELDS=(135,6,BI)
```

As shown above example, the input file is sorted by the field(s) specified in the SORT statement and then summed by the field(s) specified in the SUM statement. When fields are summed, only one record containing the final sum is written to the output file. The other records are deleted.

The SUM field can also be used to delete duplicates as shown below:

```
      SORT FIELDS=(10,8,CH,A)
SUM FIELDS=NONE
```

Setting the FIELDS to none on the SUM statement, allows only one record to be written out to the output file.

OUTREC and INREC Control Statements

The OUTREC and INREC control statements are used to reformat records. With OUTREC and INREC, you can:

- Delete fields.
- Reorder fields.
- Insert separators (blanks, zeros, or constants).

The OUTREC control statements are used to reformat records after they are sorted, copied, or merged.

```
SORT FIELDS=(10,8,CH,A)
OUTREC FIELDS=(10,8,50,2,110,8)
```

The INREC control statements are used to reformat records before they are sorted, copied, or merged.

```
INREC FIELDS=(10,8,150,5,125,2)
SORT FIELDS=(1,4,CH,A)
```

Cataloged Procedures

DFSORT may be accessed using the following Center cataloged procedures:

Procedure	Number and Type of Sort Work Files
SORT	Six (6) direct access files (default)
SORTD	Six (6) direct access files (default)
SORTD3	Three (3) direct access files
SORTD4	Four (4) direct access files
SORTD5	Five (5) direct access files
SORTD6	Six (6) direct access files

Interactive Procedure

Enter the following at ISPF, option 6 or the READY prompt: **%SORTVS**

References

DFSORT reference manuals are available in BookManager. See the BookManager chapter of this Handbook for more information on accessing and using BookManager.

DFSORT articles, news, tips, techniques, and examples are also available at the DFSORT/MVS url:

http://www.storage.ibm.com/storage/software/sort/srtmhome.htm

Chapter 6. File-AID

Introduction

File-Aid is a comprehensive file and data manipulation utility. Any type of file may be copied, browsed, modified or allocated using File-Aid. File-Aid may be used in batch or in interactive mode. You may edit DASD data sets with any record format up to 32K in length.

Also included in the interactive function is the formatted editor which programmers can display and edit records using their own record layouts.

Interactive File-AID

To use interactive File-Aid, select **=F** from the primary ISPF options panel. These File-AID options are available:

File-AID Options	Description
PARAMETERS	Specify ISPF and File-AID parameters
BROWSE	Display file contents
EDIT	Create or change file contents
UTILITIES	File-AID extended utilities
PRINT	Print file contents
SELECTION	Create or change selection criteria
XREF	Create or change record/layout cross reference
VIEW	View interpreted record layout
REFORMAT	Convert file from one format to another
COMPARE	Compare file contents
CHANGES	Display summary of File-AID changes

Batch File-AID

```
//job statement…..
//FA1 EXEC PGM=FILEAID
//STEPLIB DD DSN=KSYS1.KCFLAD.LOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSLIST DD SYSOUT=*
//SYSTOTAL DD SYSOUT=*
//DD05  DD DSN=input.data set,DISP=SHR
//DD05O DD DSN=output.data set,DISP=OLD
//SYSIN DD *
$$DD05 COPY
```

References

File-AID reference manuals are available in BookManager. For help accessing and using BookManager, see the BookManager chapter of this Handbook.

You may also select File-AID option **C** to read about File-Aid's newest features and how to use them. Pressing **PF1**, in File-Aid, will display a tutorial table of contents.

Chapter 7. CODE-1 PLUS

Introduction

CODE-1 PLUS is a mainframe product used to validate mailing addresses, standardize the addresses, and correct address errors. The product is also used to generate ZIP+4 and carrier route codes.

Address verification and ZIP Code expansion (ZIP+4) is required by the United States Postal Service (USPS) for postage discounts under the CASS Certification program. Bar coding Information, used for another type of postage discount, may also be extracted from the CODE-1 PLUS software.

The CODE-1 PLUS database is valid for all United States of America mailing addresses. This software does not contain mailing addresses for foreign countries.

CODE-1 PLUS is available in the batch environment only.

Sample CODE-1 PLUS JCL

```
//job statement …
//CODE1P  EXEC C1BM00,REGION=6M
//STEPLIB  DD DSN=KCCGCS.CODE1P.R250.LOADLIB,DISP=SHR
//G1LICEN DD DSN=KCCGCS.V250.GROUP1.G1LICEN,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//C1BMPRM   DD *
FILEDF C1BMNAM  F 0120 12000
ADDRDF L 017 30 048 30
CS ZIP S 103 C 079 20 100 02
FILEDF C1BMCOK  F 0120 12000
SA OUT     017 30 I
CS OUT     079 20 I     100
Z5 OUT     103 C
Z4 OUT     108 -
CR OUT     114
CONTRL
HEADER
//PRNTRPT   DD DSN=YOUR.CASS.OUTFILE,DISP=SHR
//PRNTXLG   DD SYSOUT=*
```

```
//C1BMNAM   DD DSN=YOUR.INPUT.FILE,DISP=SHR
//CIBMCOK   DD DSN=YOUR.OUTPUT.FILE,DISP=SHR
//ZIPIDX   DD DSN=KCCGCS.CODE1P.R250.MAY03.ZIPIDX,DISP=SHR
//COUNTY  DD DSN=KCCGCS.CODE1P.R250.MAY03.COUNTY,
//         DISP=SHR
//LCLDB   DD DSN=KCCGCS.CODE1P.R250.MAY03.LCLDB,DISP=SHR
//DTLDB   DD DSN=KCCGCS.CODE1P.R250.MAY03.DTLDB,DISP=SHR
//CITYDB   DD DSN=KCCGCS.CODE1P.R250.MAY03.CITYDB,
//         DISP=SHR
//Z4CMSTR   DD DSN=KCCGCS.CODE1P.R250.MAY03.Z4CHNG,
//         DISP=SHR
```

CODE-1 PLUS, uses your input address file to output a new file that contains the original address information, the corrected and standardized address, and the ZIP+4/carrier route codes. This product may also be coded to output a CODE-1 PLUS return code for every address that it matched, or failed to match.

An output data set may also be allocated, which will allow you to store your CASS Certificate to DASD (define as: PS, FBA, LRECL=133, BLKSIZE=27930). Once your CASS Certificate is stored on DASD, you may print multiple copies of the certificate with any IBM print utility.

Parameters are used to define your input and output files, and to tell CODE-1 PLUS which processing options to perform. The parameter statements are positional in nature and the programmer will need a CODE-1 PLUS Users Guide to understand which parameters to utilize, and how to properly use them.

The parameter statements listed above, under the C1BMPRM DD statement, are basic statements required to execute a simple procedure. Additional parameters are documented in the CODE-1 PLUS Users Guide.

Required Parameter Statements

Parameter	Description
FILEDF	Define the input (or output) file
ADDRDF	Define the input address layout
CS ZIP	Define input city, state, and ZIP code
FILEDF	Define the output (or input) file
SA OUT	Define the standardized address output
CS OUT	Define the city and state output
Z5 OUT	Define the 5-digit output ZIP code
Z4 OUT	Define the 4-digit output ZIP+4 code
CR OUT	Define the output carrier route code
CONTRL	Required by CODE-1 PLUS but has no current function
HEADER	Define report headers (if desired)

G1LICEN DD – License file required by CODE-1 PLUS.

PRNTRPT DD - Data set name of your CASS Certificate output file.

C1BMNAM DD - Data set name of your input address file.

C1BMCOK DD – Data set name of your output file for the matched and standardized address, ZIP+4 codes, and carrier route codes.

ZIPIDX DD – VSAM master file.

COUNTY DD – VSAM master file.

LCLDB DD – VSAM master file.

DTLDB DD – VSAM master file.

CITYDB DD – VSAM master file.

Z4CMSTR DD – VSAM master file.

All remaining DD statements are for other output reports.

Of all the CODE-1 PLUS generated output, Form 3553, CASS Certificate, is the most important. This is the certificate that, when placed on file at your mail drop location (post office or mail Center), will grant your agency postage discounts. When the certificate is written to DASD, multiple copies of it can be printed. The CASS Certificate is valid for 1 year.

Other reports generated by CODE-1 PLUS are control totals of input records, records processed, match attempts, matched records, and unmatched records. These statistical results are reported based on such variables as: matched on original ZIP Code, new ZIP Code determined, no ZIP Code available, insufficient address, ZIP+4 not stored, carrier route not stored, and more. Statistics are also presented based on State and the first three digits of the valid ZIP Code.

References

Customers may purchase a CODE-1 PLUS Users Manual (IBM Mainframe Series) from the following address:

Group 1 Software, Inc.
Technical Services Department Administrator
4200 Parliament Place
Suite 600
Lanham, MD 20706-1844

1-800-367-6950 for Mainframe Customer Support Hotline

Chapter 8. PKZIP/MVS

Introduction

PKZIP/MVS is a high performance data compression software product consisting of two programs, which allow file compression and decompression in an MVS environment and provides compatibility with PKZIP software running on other platforms.

PKZIP/MVS Programs

PKZIP
PKUNZIP

Both of these programs are controlled by options that allow a variety of functions to be performed.

PKZIP Functions

1. The ability to compress selected sequential files or PDS members into a single ZIP archive.
2. The ability to translate the data to be compressed between EBCDIC and ASCII.
3. The ability to create a ZIP archive file.
4. The ability to convert record oriented data, used on MVS systems, into the stream-oriented data found on a PC or UNIX system.
5. The ability to encrypt a data set into a ZIP archive during the compression process.

PKUNZIP Functions

1. The ability to decompress all or selected members from a ZIP archive file into sequential files or PDS members.
2. The ability to translate decompressed data between EBCDIC and ASCII.
3. The ability to convert stream oriented data, used on PC or UNIX systems to the record oriented data used on MVS systems.
4. The ability to view or test the contents of ZIP archives.
5. The ability to decrypt data that has been encrypted into a ZIP archive.

ZIP Archive File

ZIP Archive files hold one or more files, which have been compressed by the PKZIP software. A ZIP archive can hold up to 65,535 files, which may be compressed up to 99% of their original size. Each Archive may contain up to 4 gigabytes of data. A ZIP

archive can be transferred from one platform to another, and can be decompressed or modified by using a compatible version of the PKUNZIP program on that platform.

A ZIP Archive file holds files internally in one of the following formats listed below. If you do not specify a format then PKZIP will determine the format based on a sample of the file contents.

BINARY
TEXT
ZDW

BINARY Format files within the ZIP archive are used to hold binary data. Binary files compressed by PKZIP for MVS are exactly as held in the MVS files. No translation of the data is performed. No record delimiters are can be held within binary format files since they cannot be distinguished from the data. This means that when a binary file is decompressed, it will be decompressed at the maximum record size that the output file will support.

ZDW format files are used for binary files with record delimiters within a ZIP archive to preserve the record format. The ZDW or Zip Descriptor Word contains the exact size of a record for each record. This format is not supported by all platforms and may result in incorrect decompression of the files.

TEXT format files is used to hold data that are in textual format. Text files are always in ASCII format and will normally have embedded delimiters at the end of each record. PKZIP for MVS will translate each character in text from EBCDIC to ASCII character format. PKUNZIP will translate characters within text format files from ASCII to EBCDIC as they are decompressed.

ZIP File Attributes

A ZIP Archive file may be a sequential file on tape or disk, PDS members, or a VSAM Cluster (ESDS) file. Sequential files may be in U (Undefined), Variable (V, VB) or Fixed (F, FB, FBS) formats. PDS files must be in U (Undefined) or Fixed (F, FB) formats. PKZIP will work with any block size, but is more efficient with a larger block size. The default is 6160. The archive file must exist on MVS before a transfer from another platform can take place. The archive file can be allocated in your JCL if the file compression is to take place on the MVS platform.

PKZIP Commands

PKZIP software consists of a group of commands that are used in conjunction with the PKZIP and PKUNZIP programs. These commands are usually preceded by a hyphen "-" and allow PKZIP to perform various actions based on the commands used. See the PKZIP/MVS User Guide for a list of valid commands.

Sample JCL

Your batch JCL should contain the following statements for both, PKZIP and PKUNZIP:

- STEPLIB References the Library where PKZIP executable modules are located.
- SYSPRINT Contains the output messages
- SYSIN (Optional) Contains parameters and options to be used by PKZIP.

Example 1: PKZIP With a PDS as input –

```
//job statement…
//ZIPSTEP  EXEC PGM=PKZIP,REGION=8M
//STEPLIB  DD DSN=KSYS1.KCCZIP.LOADLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
-ZIPCUR(YOUR.SOURCE.PDS)
-ARCHIVE(YOUR.ZIP.FILE)
-ARCHUNIT(STORE)
-ARCHPRIMARY(15)
-ARCHSECONDARY(10)
-ARCHSPACE(TRK)
-ARCHTYPE(U)
-ARCHBLKSIZ(6160)
-NOPATH
 -TEXT
 -FTRAN(ASCII)
 -TRAN(ASCII)
 /*
```

The JCL above allocates the archive file "YOUR.ZIP.FILE" with the following parameters:

The -ARCHUNIT command specifies the unit on which the archive file will be allocated. The unit has been set to STORE.

The -ARCHPRIMARY command sets the primary allocation to 15.

The -ARCHSECONDARY command sets the secondary allocation to 10 units.

The -ARCHSPACE command sets the type of data units to TRACKS.

The -ARCHTYPE command sets the record format to U (undefined).

The -ARCHBLKSIZ command sets the lrecl and block size to 6160.

The -NOPATH is used to specify that only the last component of the data set name will used as the file name in the archive file.

The data set "YOUR.SOURCE.PDS" referenced by the command -ZIPCUR will be selected for compression and all members will be placed in the archive file "YOUR.ZIP.FILE".

The -FTRAN and -TRAN commands are used to specify a translation table for use when translating data from EBCDIC to ASCII.

The -TEXT command is used to specify that the selected file(s) will be compressed as a TEXT file.

Example 2: PKZIP With a PDS as input –

```
//job statement…
//ZIPSTEP  EXEC PGM=PKZIP,REGION=8M
//STEPLIB  DD DSN=KSYS1.KCCZIP.LOADLIB,DISP=SHR
//INPUT    DD DSN=YOUR.SOURCE.PDS,DISP=SHR
//ZIPOUT   DD DSN=YOUR.ZIP.FILE,DISP=(NEW,CATLG,DELETE),
//            UNIT=STORE,SPACE=(TRK,(30,15)),
//            DCB=(BLKSIZE=6160,RECFM=U,DSORG=PS)
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
-INDD(INPUT)
-ARCHIVE_OUTDD(ZIPOUT)
-NOPATH
-TEXT
-FTRAN(ASCII)
-TRAN(ASCII)
 /*
```

The JCL above allocates the archive file "YOUR.ZIP.FILE" in the JCL instead of using the PKZIP commands. The –ARCHIVE_OUTDD command is used to reference the DDNAME ZIPOUT.

The Source PDS file is also located in the JCL and the DDNAME INPUT is referenced through the use of the –INDD command.

The -NOPATH is used to specify that only the last component of the data set name will used as the file name in the archive file.

The -FTRAN and -TRAN commands are used to specify a translation table for use when translating data from EBCDIC to ASCII.

The -TEXT command is used to specify that the selected file(s) will be compressed as a TEXT file.

Example 3: PKZIP with a sequential file as input -

```
//job statement…
//ZIPSTEP  EXEC PGM=PKZIP,REGION=8M
//STEPLIB  DD DSN=KSYS1.KCCZIP.LOADLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
YOUR.SEQUENTIAL.FILE
-ARCHIVE(YOUR.ZIP.FILE)
-ARCHUNIT(STORE)
```

```
-ARCHPRIMARY(15)
-ARCHSECONDARY(10)
-ARCHSPACE(TRK)
-ARCHTYPE(U)
-ARCHBLKSIZ(6160)
-ADD
-BINARY
/*
```

The JCL above allocates the archive file "YOUR.ZIP.FILE" with the following parameters:

The -ARCHUNIT command specifies the unit on which the archive file will be allocated.

The unit has been set to STORE.

The -ARCHPRIMARY command sets the primary allocation to 15.

The -ARCHSECONDARY command sets the secondary allocation to 10 units.

The -ARCHSPACE command sets the type of data units to TRACKS.

The -ARCHTYPE command sets the record format to U (undefined).

The -ARCHBLKSIZ command sets the lrecl and block size to 6160.

The sequential file "YOUR.SEQUENTIAL.FILE" will be selected for compression and placed in the archive file "YOUR.ZIP.FILE" in binary format as specified by the -BINARY command. If your file is a text file, remove the -BINARY command from the JCL and replace with the -TEXT command. You will also want to add the -FTRAN and -TRAN commands since they are used to specify a translation table when translating text data from EBCDIC (MVS character set) to ASCII (character set used within the archive file).

The -ADD command is used to add files when creating a new archive file.

The -UPDATE command should be used instead when adding or updating an existing archive file.

Example 1: PKUNZIP using a PDS for Extraction -

```
//job statement…
//UNZIPSTP EXEC PGM=PKUNZIP,REGION=8M
//STEPLIB  DD DSN=KSYS1.KCCZIP.LOADLIB,DISP=SHR
//INPUT    DD DSN=YOUR.ZIP.FILE,DISP=SHR
//OUTFLE   DD DSN=YOUR.OUTPUT.PDS,DISP=(,CATLG,DELETE),
//            UNIT=STORE,SPACE=(TRK,(15,10,10)),
//             DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB)
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
```

```
-ARCHINDD(INPUT)
-OUTDD(OUTFLE)
-TEXT
-FTRAN(ASCII)
-TRAN(ASCII)
/*
```

In the JCL example above, the –OUTDD command references the DDNAME OUTFLE, in which the extracted members will be placed.

The -ARCHINDD command references the DDNAME INPUT defining the archive file "YOUR.ZIP.FILE".

The -FTRAN and -TRAN commands are used to specify the translation table for use when extracting text files from an archive file.

The -TEXT command is used to specify that the selected file(s) will be extracted as a text file(s).

Example 2: PKUNZIP using a sequential file for extraction -

```
//job statement…
//UNZIPSTP EXEC PGM=PKUNZIP,REGION=8M
//STEPLIB  DD DSN=KSYS1.KCCZIP.LOADLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
-ARCHIVE(YOUR.ZIP.FILE)
-OUTUNIT(STORE)
-OUTPRIMARY(15)
-OUTSECONDARY(10)
-OUTSPACE(TRK)
-OUTTYPE(FB)
-OUTLRL(80)
-OUTBLKSIZ(80)
-HLQ(YOUR.SEQUENTIAL.FILE/YOUR.SEQUENTIAL.OUTPUT)
-BINARY
/*
```

In the JCL example above the -ARCHIVE command references the archive file "YOUR.ZIP.FILE".

The following commands are used to create the sequential output file "YOUR.SEQUENTIAL.OUTPUT":

The -OUTUNIT command specifies the unit on which the sequential file will be allocated. The unit has been to STORE.

The -OUTPRIMARY command sets the primary allocation of the file to 15 units.

The -OUTSECONDARY command sets the secondary allocation to 10 units.

The -OUTSPACE command sets the type of data units to TRACKS.

The -OUTTYPE command sets the record format to FB (Fixed Block).

The -OUTLRL command sets the logical record length to 80.

The -OUTBLKSIZ command sets block size to 80.

The -HLQ command specifies the name of the file to be extracted from the archive file, followed by a slash, and then the name of the new sequential output file.

Our file "YOUR.SEQUENTIAL.FILE" will be selected for extraction and will be placed in the newly created file "YOUR.SEQUENTIAL.OUTPUT".

The -BINARY command is used to specify a binary file. If your file is a text file, remove The -BINARY command from the JCL and replace with the –TEXT command.

You will also want to add the -FTRAN -TRAN commands since they are used to specify a translation table when translating text data from EBCDIC (MVS character set) to ASCII (character set used within the archive file).

User Requirements

- PKZIP/MVS is only a compression/decompression software product. It does not provide a means to transfer your compressed data from one platform to another.

- Binary file specification must be used when transferring the ZIP archive file.

- You cannot combine sequential data sets and PDS members for compression or extraction in the same execution of PKZIP software.

- PKZIP/MVS requires additional licensing of GZIP is required to compress or decompress files that are 4 gigabytes or larger.

- PKZIP/MVS encryption provided with this release is limited to 40 bits. This security level is not intended to protect sensitive data. Additional Software licensing is required for greater encryption protection.

References

PKZIP manuals are available from PKWARE, Inc.:

PKWARE, Inc.
9009 Springboro Pike
Miamisburg, Ohio 45342
(937) 847-2374

PKZIP manuals can also be viewed online at *www.pkware.com/support.*

Chapter 9. HOURGLASS 2000

Introduction

HourGlass 2000 is a mainframe software product that will allow you to temporarily alter the system date and/or time to any date in the future (up to September 17, 2042). By using HourGlass 2000 to test your batch applications, you can determine, from a program perspective, the impact of a future run date.

User Requirements

The key to using HourGlass 2000 is in understanding the two DD statements. One DD statement is used to change the computer system's run date to a test date in the future. The second DD statement is used to change the computer system's run time to another time of the day or night.

Date DD Statement

To execute a batch application test using HourGlass 2000, you must utilize a special DD statement in each step of your process. It is through the use of the date DD statement that you can select the century, the year, and the Julian date that you wish to process with.

Date DD Statement Format

//HGcyyddd DD DUMMY
HG - calls the HourGlass 2000 procedure
c - is the century indicator
yy - is the year indicator
ddd - is the Julian day indicator.

Sample Date Setting DD's

January 1, 2000

//HG100001 DD DUMMY
HG - calls the HourGlass 2000 procedure
1 - specifies the 21st century.
00 - specifies year 00 (of the 21st century)
001 - specifies Julian day 001 (January 1st)

February 1, 1995

//HG095032 DD DUMMY	
HG	- calls the HourGlass 2000 procedure
0	- specifies the 20th century.
95	- specifies year 95 (of the 20th century)
032	- specifies Julian day 032 (February 1st)

July 4, 2041

//HG141185 DD DUMMY	
HG	- calls the HourGlass 2000 procedure
1	- specifies the 21st century.
41	- specifies year 41 (of the 21st century)
185	- specifies Julian day 185 (July 4th)

November 30, 1998

//HG098334 DD DUMMY	
HG	- calls the HourGlass 2000 procedure
0	- specifies the 20th century.
98	- specifies year 98 (of the 20th century)
334	- specifies Julian day 334 (November 30th)

A date DD statement must be placed in each step of your test JCL to perform the date change. Please note that on leap years, the Julian date will differ for the months of March through December.

Also, the century indicator uses a "0" for the 20th century (19xx), and a "1" for the 21st century (20xx).

Online HourGlass Aids

There is online aid available that will generate the date DD statement for you and place it in your JCL.

From a command line within an edit session, key HGDD. Place an "A"fter or "B"efore on the proper line number based on where you want to have the DD statement inserted. After this, another panel will display where you are prompted to insert the date.

You may also choose to enter a time value. If not, a dummy time statement will be generated for you. After you have entered the date, press ENTER and the statements will be inserted.

You can pass the desired date directly to the CLIST in the format: ccyy-mm-dd (where ccyy is the year, mm is the month, and dd is the day of the month).

For example, to pass July 4, 2041 to the DD statement, you would key in hgdd 2041-07-04. This will automatically generate the your statements without going to the above mentioned panel.

There is an online aid available for generating the date DD statement in the proper format for you. From the command line in TSO/ISPF, key **TSO %HGDDNAME** to call the online CLIST .

Reply to the date query in the format: ccyy-mm-dd (where ccyy is the year, mm is the month, and dd is the day of the month). To set the proper date DD statement format for July 4, 2041, key in 2041-07-04.

The CLIST will respond with "DD statement to use: //HG141185 DD DUMMY". After you exit the online CLIST (by pressing PF3), code this date DD statement into each required job step in your JCL.

Time DD Statement

HourGlass 2000 can alter the run time of your application test on a step-by-step basis. Through the use of the time DD statement, you can adjust the computer system's run time, forward or backward, for a maximum of 23 hours and 59 minutes in either direction.

The format for the Time DD statement is:

//HGdhhmm DD DUMMY
HG - calls the HourGlass 2000 procedure
d - is the direction indicator (forward/backward)
hh - is the number of hours to offset the time
mm - is the number of minutes to offset the time

The direction indicator ("d") is used to determine if you wish to add hours and/or minutes to the current time or, if you wish, to subtract hours and/or minutes from the current time. HourGlass 2000 will accept a "P" (plus) or "E" (East) as the direction indicator when you add hours and/or minutes to the current time.

HourGlass 2000 will also accept "M" (minus) or "W" (West) as the direction indicator when you subtract hours and/or minutes from the current time.

The hours ("hh") and minutes ("mm") indicators are strictly numeric, with values of 00 to 23 for the hours, and 00 to 59 for the minutes.

Sample Time Setting DD

Add 2 Hours and 30 Minutes to System Time

//HGP0230 DD DUMMY	
HG	- calls the HourGlass 2000 procedure
P	- will add the current time (P = plus)
02	- will adjust current time by 02 hours
30	- will adjust current time by 30 minutes

Subtract 3 hours and 0 Minutes from System Time

//HGW0300 DD DUMMY	
HG	- calls the HourGlass 2000 procedure
W	- will subtract from the current time (W = West, or in this case, three zones West of your location.
03	- will adjust current time by 03 hours
00	- will adjust current time by 00 minutes

As with the date DD statement, you must place the time DD statement in each step of your test JCL that you wish to deviate from the current system time. You can also include both the date DD statement and the time DD statement in the same job step, when you want to alter both the date and time from the current system date/time.

Job Statement Date and Time Setting

You can specify Hourglass 2000 at the job level by modifying your JOB statement's "Programmer Name Field". If both date & time are specified, separate them by a space. You can override this by using an Hourglass Step DD statement.

Example altering Date and Time:

//TESTXX JOB(ACCOUNT,XXXX,1),'//HG098015 //HGP0100',ETC.

Example altering Date Only:

//TESTXX JOB(ACCOUNT,XXXX,1),'//HG098015',ETC.

Testing HOURGLASS 2000 with C/370

To test with C/370, modify your JCL to concatenate a new data set in front of your current loadlib.

Example:

```
//STEP1 EXEC PGM=CPROG11
//STEPLIB DD DSN=KCUSER.HGLTEST.LOAD,DISP=SHR ←new loadlib for HG
//       DD DSN=YOUR.PROD.LOAD,DISP=SHR     ← your loadlib
//SYSPRINT DD SYSOUT=*
//SYSTERM DD SYSOUT=*
//HG102074 DD DUMMY          ←HOURGLASS DATE OF 2002-03-15
```

Testing HOURGLASS 2000 with LE/370 or IBM COBOL

To test LE/370 or IBM COBOL, you will need to modify your JCL to concatenate a new data set in front of your current loadlib.

Example:

```
//SMPPGM3 EXEC PGM=CBLLOCT,REGION=4M
//STEPLIB  DD DSN=KCUSER.HGLTEST.LOAD,DISP=SHR <-new loadlib for HG
//        DD DSN=YOUR.TEST.LOADLIB,DISP=SHR     ← your loadlib
//        DD DSN=SYS1.LE.PROD.SCEERUN,DISP=SHR  ←LE/370 loadlib
//DATA    DD DSN=YOUR.INPUT.DATA,DISP=SHR
//SYSOUT   DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSUDUMP  DD SYSOUT=*
//HG100060 DD DUMMY          HOURGLASS DATE OF 2000-02-29
/*
```

Testing HOURGLASS 2000 with ASSEMBLER

To test assembler applications that use the STCK assembler instruction, you will need to concatenate 'KCUSER.SAMPLE.CNTL(STCK)' in front of your source code on the SYSIN DDname.

You will also need to add a "COPY STCK" statement to the beginning of the assemble SYSIN. This will cause each occurrence of the STCK instruction will be converted to a format of SVC 11, which returns the TOD value, just like the STCK instruction so HG2000 will be able to intercept it. Beware, however, that the use of the SVC causes registers 0, 1, 14, and 15 to be altered. If your application requires that these registers remain intact, you will need to concatenate 'KCUSER.SAMPLE.CNTL(STCKSAVE)' this contains a non-reentrant version of the replacement macro that saves and restores these registers so that no register alteration is done. This Macro generates non-reentrant code to save registers 0, 1, 14, and 15. You must use GETMAIN/FREEMAIN to make this code reentrant.

```
//ASM.SYSIN  DD DSN=KCUSER.SAMPLE.CNTL(STCK),DISP=SHR
//          DD DSN=YOUR.BAL.SOURCE,DISP=SHR
```

Testing HOURGLASS 2000 with DB2

To test DB2 applications which use the SQL special registers CURRENT DATE, CURRENT TIME, CURRENT TIMESTAMP as well as default values on INSERTs for DATE, TIME, and TIMESTAMP data type, you will need to contact the Customer Support Center at (816) 926-6681.

The NITC staff will need to make modifications to each of your DB2 subsystems you wish to test this option with. After these modifications are made, you will just need to insert the proper HourGlass statement, as described above, into your batch job.

Chapter 1. VS FORTRAN

Introduction

VS FORTRAN is a programming language best suited to applications that involve mathematical computations and other manipulation of arithmetic data. The VS FORTRAN product consists of a compiler, a run-time library of subprograms, and an interactive debugging facility.

Usage

Under MVS, you process a VS Fortran program by submitting batch jobs to the operating system. These batch jobs may be submitted using foreground processing (ISPF option, **=4.3**) or background processing using cataloged procedures.

See the Program Creation and Execution chapter of this Handbook if you would like more information on program creation, compilation and execution.

Cataloged Procedures

Procedure	Description
VSF2C	Compile
VSF2CL	Compile and link edit
VSF2CLG	Compile, link edit and execute
VSF2LG	Link edit and execute
VSF2G	Execute only
VSF2CG	Compile and execute
VSF2L	Link edit only

Symbolic Parameters

Parameter	Default	Description	Used in Procs
FVPGM	FORTVS2	COMPILER NAME	VSF2C VSF2CG VSF2CL VSF2CLG
FVREGN	2100K	FORT STEP REGION	VSF2C VSF2CG VSF2CL

			VSF2CLG
FVPDECK	NODECK	COMPILER DECK	VSF2C VSF2CG VSF2CL VSF2CLG
FVPOLST	NOLIST	COMPILER LIST OPTION	VSF2C VSF2CG VSF2CL VSF2CLG
FVPOPT	0	COMPILER OPTIMIZATION	VSF2C VSF2CG VSF2CL VSF2CLG
FVTERM	SYSOUT=*	FORT.SYSTERM OPERAND	VSF2C VSF2CG VSF2CL VSF2CLG
FVLNSPC	3200,(25,6)	FORT.SYSLIN SPACE	VSF2C VSF2CG VSF2CL VSF2CLG
GOF5DD	DDNAME=SYSIN	GO.FT05F001 OPERAND	VSF2CG VSF2CLG VSF2G VSF2L VSF2LG
GOF6DD	SYSOUT=*	GO.FT06F001 OPERAND	VSF2CG VSF2CLG VSF2G VSF2L VSF2LG
GOF7DD	SYSOUT=B	GO.FT07F001 OPERAND	VSF2CG VSF2CLG VSF2G VSF2L VSF2LG
GOREGN	100K	GO STEP REGION	VSF2CG VSF2CLG VSF2G VSF2LG
PGMLIB	&&GOSET	LKED.SYSLMOD DSNAME	VSF2CL
PGMNAME	MAIN	LKED.SYSLMOD MEMBER NAME	VSF2CL
GOPGM	MAIN	PROGRAM NAME	VSF2G VSF2LG
LKLNDD	DDNAME=LOADIN	GO.FT05F001 OPERAND	VSF2L VSF2LG

JCL Examples

Compile -

```
//job statement…
//STEP1    EXEC VSF2C,PARM='SDUMP'
//FORT.SYSLIN DD DSN=object.dataset(member),DISP=SHR,UNIT=
//FORT.SYSPRINT DD SYSOUT=*
//FORT.SYSTERM  DD SYSOUT=*
```

Compile and Execute

```
//job statement…
//STEP1    EXEC VSF2CG,PARM='SDUMP'
//FORT.SYSPRINT DD SYSOUT=*
//FORT.SYSTERM  DD SYSOUT=*
//SYSIN    DD DSN=fortran.program(member),DISP=SHR
//GO.SYSLOUT  DD SYSOUT=*
//GO.FT04F001 DD SYSOUT=*
//GO.FT05F001 DD SYSOUT=*
//GO.FT06F001 DD SYSOUT=*
```

Compile, Link and Execute

```
//job statement…
//STEP1    EXEC VSF2CLG,PARM='SDUMP'
//FORT.SYSLIN DD DSN=object.dataset(member),DISP=SHR,UNIT=
//FORT.SYSPRINT DD SYSOUT=*
//FORT.SYSTERM  DD SYSOUT=*
//SYSIN    DD DSN=fortran.program(member),DISP=SHR
//LKED.SYSLMOD DD DSN=load.dataset(member),DISP=SHR,
//     UNIT=
//LKED.SYSLIN DD DSN=object.dataset(member),DISP=SHR
//LKED.SYSPRINT DD SYSOUT=*
//GO.SYSLIN DD DSN=load.dataset(member),DISP=SHR
//GO.FT04F001 DD SYSOUT=*
//GO.FT05F001 DD SYSOUT=*
//GO.FT06F001 DD SYSOUT=*
```

References

Online documentation for VS Fortran is available using BookManager. For help accessing or using BookManager, see the BookManager chapter of this handbook.

Chapter 2. PL/I

Introduction

PL/I is a high-level language that combines the scientific capabilities of FORTRAN, the bit manipulation capabilities of ASSEMBLER, and the business-type application capabilities of COBOL.

Usage

Under MVS, you process a PL/I program by submitting batch jobs to the operating system. These batch jobs may be submitted using foreground processing (ISPF option, **=4.5**) or background processing using cataloged procedures.

See the Program Creation and Execution chapter of this Handbook if you would like more information on program creation, compilation and execution.

Cataloged Procedures

Procedure	Description
PLIXC	Compile
PLIXCG	Compile and execute
PLIXCL	Compile and link
PLIXCLG	Compile, link and execute
PLIXG	Execute only
PLIXLG	Load and execute

References

Online documentation for PL/I is available using IBM's BookManager. For help accessing or using BookManager, see the BookManager chapter of this handbook.

Chapter 3. IBM COBOL

Introduction

This chapter documents how to compile and link COBOL programs using IBM COBOL. The procedures provided in this chapter will always use the most current version of IBM COBOL on our mainframes so you no longer need to be concerned with what version of COBOL you are using. However, if you are still using OS/VS COBOL or COBOL II the Center recommends they be converted to the current version of IBM COBOL.

Usage

IBM COBOL programs are compiled and link edited by submitting batch jobs to the operating system. These batch jobs may be submitted using foreground processing (ISPF option, **=4.17**) or background processing using cataloged procedures.

See the Program Creation and Execution chapter of this Handbook if you would like more information on program creation, compilation and execution.

Cataloged Procedures

Procedure	Description
IGYWC	Compile
IGYWCG	Compile, load and execute
IGYWCL	Compile and link edit
IGWCLG	Compile, link edit and execute
IGYWPL	Prelink and link edit
IGYWCPL	Compile, prelink, and link edit
IGYWCPLG	Compile, prelink, link edit and execute

JCL Example

Compile and Link Edit -

```
//job statement…
//STEPALL EXEC IGYWCL,PARM.COBOL=('MAP,LIST,OBJECT')
//COBOL.SYSIN    DD
DSN=KIM999.GENERAL.COBOL(IGZPGM),DISP=SHR
//LKED.SYSLMOD DD DSN=KIM999.GENERAL.LOAD(IGYPGM),
// DISP=SHR,UNIT=
/*
```

References

Online COBOL documentation is available using IBM's BookManager. For help accessing or using BookManager, see the BookManager chapter of this Handbook.

Chapter 4. CA-MetaCOBOL

Introduction

CA-MetaCOBOL is a software system that assists programmers in writing, testing, debugging, optimizing and converting COBOL source programs through the use of conditional macro procedures. The CA-MetaCOBOL Translator provides procedures to perform specific operations during virtually all aspects of program implementation and maintenance.

Cataloged Procedures

Procedure	Description
MTACBL	Basic MetaCOBOL procedures
MTACL	Cobol/MetaCOBOL Compile and Link
MTACNLN	Conversion and Level renumbering
MTAEFST	Efficiency and Standards
MTAPNB	Paragraph renumbering

References

To order the manuals, write or call:

Computer Associates InterNational
2291 Woodoak Drive
Herndon, VA 20171

Attn: Federal Division

(703) 709-4500

Please indicate that CA-MetaCOBOL is installed at the National Information Technology Center to allow for direct ordering and billing.

Chapter 5. CA-Optimizer II

Introduction

CA-Optimizer II is a COBOL programmer productivity product for the MVS/ESA environment. CA-Optimizer II automatically optimizes COBOL programs and assists programmers in creating, testing, debugging, fine-tuning, and maintaining COBOL programs.

Features

- CA-Optimizer – increases the efficiency of a program and greatly enhances the source listing.
- Detector – aids in debugging a program in the production and test environments.
- Analyzer – allows the programmer to thoroughly fine-tune the program and assure its quality.

Management Reporting System (MRS)

Installed with CA-Optimizer II is the management reporting system (MRS). MRS analyzes any standard IBM load module library but does not support concatenated load library data sets. Specify each load library to be analyzed in a separate MRS execution run. MRS distinguishes between data and procedure bytes. The data bytes include only the areas in the program generated at compile time, and does not include data areas acquired dynamically at execution time for a program compiled with the RENT option.

Cataloged Procedures

Procedure	Description
CAIMVSII	MVS COBOL/Optimizer II Compile and Link Editor
CAIARII	Optimizer II Reporter
CAIMRSII	Optimizer II Management Reporting System (MRS)
CASTRII	Optimizer II Statistical Reporting System

Using CA-Optimizer II with CA-IDMS/DC

CA-Optimizer II works with BATCH-DC programs. Output from the IDMSDMLC compiler should be input to the CA-Optimizer II compile procedure. Option IDMSDC is required for BATCH-DC programs.

```
//STEP2   EXEC  CAIMVSII
//   COBOPT='IDMSDC…….'
```

References

CA-Optimizer manuals may be ordered from:

Computer Associates InterNational
2291 Woodoak Drive
Herndon, VA 20171

Attn: Federal Division

(703) 709-4500

Addition reference information may also be obtained by accessing url: http://www.ca.com/ and entering in the SEARCH box: **CA-Optimizer**.

Chapter 6. CA-Librarian

Introduction

CA-Librarian is a generalized data storage and retrieval system designed to assist in the development and maintenance of source programs and other set a data records. System facilities prohibit damage, loss or unauthorized access to master files.

Definitions

A CA-Librarian master file consists of modules in a direct file or a sequential file. It is not a partitioned data set.

Modules contain documentation of the data records and control information pertaining to a single module.

A module can be 80 character records, or print records of 121 or 133 characters with carriage control characters.

Batch JCL

Initializing a Disk Master File -

```
//        EXEC LIBRAREN,REGION=128K,PARM='NJTA,NRJS'
//MASTER   DD DISP=(NEW,CATLG),DSN=ANYNAME,UNIT=STORE,
//           SPACE=((CYL, or TRK),(NUMBER OF CYL or TRK)),
//           DCB=(DSORG=DA,BLKSIZE= * bbb)
//SYSPRINT DD SYSOUT=A,DCB=(RECFM=FBA,BLKSIZE=3325)
//OSJOB    DD DUMMY
//SYSIN    DD *
-OPT INIT,DISK,SEQ=COBOL,NOBYPP
/*
```

BLKSIZE - (select one of the following) **bbb**= (3390, 27998, 18452, 13682, 7548, 4136, 2942, 2082, 1086)

-OPT INIT,DISK is the minimum required. Additional options can be specified on the -OPT statement and are listed below.

ARC=LEVELS Specifies that archived modules are permitted on the master and designates the maximum number of archiving levels retained. From 2 to 255 levels. The ARC option does not set a master file default; you must specify ARC on the -ADD or -SEL module control statement for each module to be archived.

BKUPINT=NN An automatic backup will be taken every NN runs against the master file. Specified at initialization only.

BVS1=VOLSER AND BVS2=VOLSER Specify the volume serial numbers of tapes for the automatic backup. Specified at initialization only.

COMPRESS=FULL, PART, OR NONE A trade off between saving space or execution time. (initialization only).

EXEC **Creates** output to OSJOB.

LIST List contents of a module.

NOBYPP Used if customer does not want to bypass the password (initialization only).

NOGPO Inhibits the Group Processing Option. If NO specified the Group Processing Option is available. This default cannot be changed.

NOPR Instructs the CA-Librarian to print on the Update Record listing only the first and last records when two or more consecutive records are deleted from a module on this master file. Default is all records are printed. Can also be specified on the -SEL.

NORESEQ Suppress automatic resequencing of modules.

NOSEP Suppress page eject.

SEQ= | /71,8,10,10/

| /A, B, C, D/ "A" is the beginning column, "B" is the length, "C" is the increment, and "D" is the starting value.

| COBOL

NOSEQUPD If sequenced updating not wanted. (IEBUPDTE uses sequenced updating).

VAR Invokes the CA-Librarian Source-Load Audit Trail facility for every module on the master file. It is activated when the module is selected for execution via the EXEC option. VAR or NOVAR can be coded on the -ADD, -OPT, or -SEL statement to override the default for a single execution.

For more information on previously listed options, please see the CA-Librarian Command Reference Manual (Batch). This manual is available in BookManager. See the References section of this chapter.

Running Against Disk Master Files -

```
//STEP EXEC LIBRAREN,PARM='NRJS,NJTA',
//      SRCLIB=FULLY.QUALIFIED.NAME
//OSJOB DD DUMMY
//SYSIN DD *
```

```
(CA-Librarian Control Statements and Data)
/*
```

OSJOB DD &&LIBSRC This data set is passed and can be used as SYSIN to a compiler if EXEC is coded on the -OPT or -SEL statement in this CA-Librarian step.

```
//STEP EXEC LIBRAREN,PARM='NRJS,NJTA',
//      SRCLIB=FULLY.QUALIFIED.NAME
//OSJOB DD DUMMY
//SYSIN DD *
  (CA-Librarian Control Statements and Data)
//STEP2 EXEC LIBRAREN,PARM='NRJS,NJTA',
//       SRCLIB=FULLY.QUALIFIED.NAME
//SYSIN DD DSN=&&LIBSRC,DISP=(OLD,DELETE),
//   DCB=BLKSIZE=1680
/*
```

* SRCLIB = Name of master file

Back Up Disk Master File

A counter set when the file is created provided the parameter BKUPINT initiates automatic backup=NN has been specified on the -OPT statement when the master file is initialized.

A backup may be taken at any time by specifying -OPT BKUPTAPE.

If BVS1 and BVS2 are used the DSNAMES are DSVOLSER and DSVOLSER.

At NITC, it is preferable that a generation data set be used for backup. This would cause a non-specific output volume to be asked for by the operating system. Using cycle control a specific volume request would be made, which requires an operator to find it in the library before a customer's job can continue.

JCL for using generation data set for backups -

```
//       EXEC LIBRAREN,REGION=128K,PARM='NJTA,NRJS'
//MASTER   DD DISP=OLD,DSN=FULLY.QUALIFIED.NAME
//SYSPRINT DD SYSOUT=A,DCB=(RECFM=FBA,BLKSIZE=3325)
//INDEX    DD SYSOUT=A,DCB=(RECFM=FBA,BLKSIZE=3325)
//LIST     DD SYSOUT=A,DCB=(RECFM=FBA,BLKSIZE=3325)
//SYSPUNCH DD DUMMY
//OSJOB    DD DUMMY
//BACKUP   DD UNIT=(TAPE,,DEFER),DISP=(,CATLG,DELETE),
//         DSN=GENERATION.DATA.GROUP.NAME(+1)
//SYSIN    DD *
-OPT BKUPTAPE
-END
/*
```

Index can also be specified on the -OPT statement if you wish to create an index of the master file for immediate reference to the contents of the backup.

To restore the latest backup use -OPT as the SYSIN and the following DD statement for the backup file:

```
//BACKUP   DD UNIT=(TAPE,,DEFER),DISP=(,KEEP),
//            DSN=GENERATION.DATA.GROUP.NAME(+0)
```

For more batch JCL information, see the Librarian Command Reference Manual (Batch). This manual is available in BookManager.

Control Statements, Commands and Processing Options (BATCH)

Control statements activate CA-Librarian operations. A CA-Librarian control statement begins with a command; it can refer to a specific module and can include processing options and customer commands.

For information on different kinds of control statements, control commands and processing options please see Librarian Command Reference Manual (Batch).

Syntax Checker for COBOL

Code SYNCHK on the -ADD or -SEL statement.

For more information on the syntax checker please see the Librarian Command Reference Manual (Batch).

ELIPS

ELIPS (Extended Librarian Interactive Productivity Services) is an ISPF/PDF application that provides complete CA-Librarian access and update capabilities to the online customer.

The following functions are available through ISPF option **=D.L.E**

ArCLIST	**Browse**	**Print**
Delete	**Edit**	**Utility**
Info	**Options**	**Copy**
Rename	**Types**	

For more information on functions and options available through ELIPS, see the Librarian Command Reference Manual (ELIPS). This manual is available in BookManager.

Information is also available through the online tutorial that can be accessed by keying help on the ELIPS panel command line. To exit the Help facility, enter END.

Copy/Move Utility

CA-Librarian copy/move utility is available through (D.L.E). Through this utility the online customer can move or copy CA-Librarian modules from one master CA-Librarian to another.

For more information on functions available through ELIPS please see the Librarian Command Reference Manual (ELIPS). Information is also available through the online tutorial that can be accessed by entering help on the ELIPS panel command line.

LIBAUDIT

LIBAUDIT compares prior versions to new versions and stores only the differences with the original. LIBAUDIT can recreate any program version online, at any time. LIBAUDIT automatically keeps track of every statement that has ever been included in a module, thereby maintaining an online audit trail. LIBAUDIT saves time in batch and allows you online access to all program changes without losing your audit trail.

For more information on the LIBAUDIT please see the Librarian Command Reference Manual (Batch).

COMPARATOR II

COMPARATOR II compares sequential files, modules of a partitioned data set (PDS), or modules of a CA-Librarian master file. These files can have record formats of fixed, variable or undefined.

The following statement is needed to execute COMPARATOR II:

```
//STEPxx  EXEC  PGM=COMPRTOR
```

For more information on the COMPARATOR II please see COMPARATOR II manual.

Available Utility Programs

LIBRCOPY - Is used to copy a member from an archived master to a non-archived master or to copy a member to an archived master and update the archived member instead of replacing it.

LIBREXIT – This is the CA-Librarian output exit that inserts the SLAT variables into COBOL, Assembler, and PLI source members. The CA-Librarian LANG value is used to determine whether the member is COBOL, Assembler, or PLI. The tables for the language types are at the labels "COB," "ASM," and "PLI."

LIBRSCAN - Is a utility program for The CA-Librarian to provide a source-to-load cross-reference report. The program compares a specified load library against one or more master files of The CA-Librarian, and generates a report that lists either the

exceptions or the full results of the comparison. All source modules represented within a given load module are verified.

LOADSCAN - This program compares two load modules. It ignores differences in the CA-Librarian SLAT variables. All other differences, whether they are linkage editor options or the code itself, will be reported by LOADSCAN as differences between the two load modules.

For more information on the utility programs please see the Librarian System Services Guide.

References

Online documentation for CA-Librarian is available using IBM's BookManager. For help accessing or using BookManager, see the BookManager chapter of this handbook.

If you wish to order CA-Librarian manuals, please contact:

Computer Associates International
2291 Woodoak Drive
Herndon, VA 20171

Attn: Federal Division

(703) 709-4500

Please indicate that you use the CA-Librarian installed at USDA National Information Technology Center in Kansas City, Missouri.

Addition reference information may also be obtained by accessing url: http://www.ca.com/ and entering in the SEARCH box: **librarian**.

Chapter 7. REXX/370 Compiler

This chapter provides an overview of the features and functions of the IBM Compiler for SAA REXX/370. You may not want to compile all REXX programs. For instance, you may not want to compile simple REXX programs that are used infrequently. REXX/370 programs may be compiled interactively or in batch.

Usage

One effective way of using the Compiler to develop REXX programs is the following:

1. Compile the program with the TRACE Compiler option. This step performs comprehensive error checking and produces an output that can be traced.

2. Debug the program using the output of the previous step.

3. Compile the program with the NOTRACE Compiler option.

Interactive Compiles

REXX/370 programs may be compiled interactively by selecting ISPF option, **=D.RX**.

The following REXX/Tools panel will display:

1 COMPILE - Invoke REXX Interpretive Compiler
2 USER MANUAL - Read REXX/Tools Basic Services User Manual
3 SQL MANUAL - Read REXX/Tools SQL Services User's Guide

Select option 1 to compile your REXX program. At the next panel, enter the following:

```
Source Data Set Name   ===>  'your..REXX.EXEC(MYEXEC)'
Object Data Set Name   ===>  'your.REXX.LOAD(MYEXEC)'
Listing Data Set Name  ===>  'your.list.dataset(MYEXEC)'

Compiler Options  ===>
 (COMPRESS, ENCODE, NAME(), PAGESIZE(), VERSION(), XREF)
```

Cataloged Procedure

Procedure	Description
RXTCL	Compile and link.

JCL Example

This example compiles, links and executes a REXX exec (program) called MYEXEC. 'Mable Smith' is a parameter passed to the program.

```
//job statement …
//STEP1  EXEC RXTCL ,RXCPARM='COMPRESS XREF NAME(MYEXEC)'
//C.SYSIN   DD DSN=XXX999.GENERAL.REXX.EXEC(MYEXEC),DISP=SHR
//C.SYSPRINT DD
DSN=XXX999.GENERAL.REXX.RXLIST(MYEXEC),DISP=SHR
//L.SYSLMOD  DD
DSN=XXX999.GENERAL.REXX.LOAD(MYEXEC),DISP=SHR
//STEP2 EXEC PGM=REXXTEST,PARM='Mable Smith'
//STEPLIB DD DSN=XXX999.GENERAL.REXX.LOAD(MYEXEC),DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD DUMMY
```

References

Select options 2 and 3 of the REXX options panel for the REXX/Tools Basic Services User Manual and REXX/Tools SQL Services User's Guide, respectively.

Online documentation for REXX/370 is also available using IBM's BookManager. For help accessing or using BookManager, see the BookManager chapter of this handbook.

Chapter 8. ABEND-AID

Introduction

ABEND-AID is an operating system enhancement that transfers the task of analyzing program abends from the programmer to the computer. It intercepts an abend before an IBM dump is triggered replacing the dump, in most instances, with a formatted analysis of the problem, reducing CPU time for the dump by one-third and printed output to a minimum.

Usage

The only requirement necessary to receive ABEND-AID processing is the presence of a SYSUDUMP DD statement in your job stream. At the conclusion of ABEND-AID processing the abend is solved or unsolved. This determination is based on the type of abend and the conditions encountered by the ABEND-AID processing routines. An abend code table is consulted for each abend code, using the solved and unsolved entries to specify how the IBM dump is to be handled.

These are the available options:

1. NO DUMP - In this case the IBM dump is completely suppressed.

2. FORMATTED - For this option, the front section of the dump is allowed to continue. This consists of the normal formatting provided by IBM down to the point where the actual dump of storage begins. No storage is dumped.

3. DUMP - If this option is specified, the dump is allowed to continue.

For most solved conditions the IBM dump is suppressed. Regardless of the options specified, the customer has the capability of overriding the specification through the use of JCL overrides:

//ABNLDUMP DD DUMMY - Forces printing of normal SYSUDUMP Dump, regardless of installation options, additional to ABEND-AID output.

//ABNLNODP DD DUMMY - Tells ABEND-AID to suppress printing. The normal SYSUDUMP after ABEND-AID processing regardless of installation options.

//ABNLFMTD DD DUMMY - ABEND-AID prints only the formatted portion of the SYSUDUMP in addition

to the ABEND-AID output. Printing the contents of memory in hexadecimal is suppressed.

//ABNLHELP DD DUMMY - ABEND-AID prints the HELP output which explains the use ABEND-AID input.

//ABNLALL DD DUMMY - ABEND-AID formats all program storage and control blocks regardless of installation options.

//ABNLNONE DD DUMMY - ABEND-AID suppresses the printing of program storage areas/control blocks.

//ABNLIGNR DD DUMMY - Stops ABEND-AID processing completely for this step. The normal SYSUDUMP dump will still be printed.

//ABNLWSPT DD DUMMY - Ensures program storage for all active programs will be formatted by ABEND-AID regardless of site options.

//ABNLNWSP DD DUMMY - Suppresses the printing of ABEND-Aid's program storage for this step.

//ABNLPCBS DD DUMMY - Ensures data management control blocks are printed by ABEND-AID for this step regardless of installation options.

//ABNLNCBS DD DUMMY - Suppresses printing of ABEND-AID data management control blocks for this step.

//ABENDAID DD SYSOUT=A - ABEND-AID directs its output to the ABENDAID data set, and the full dump, required, will be written to the SYSUDUMP Sysout.

User Abend Codes

User abends are currently set to receive a full dump whenever a dump is requested. ABEND-AID can be set to produce a formatted dump only for specific user abend codes. Questions on changing the type of dump produced in response to a particular user abend code should be directed to the Information Management and Customer Support Division.

Procedures

To generate an ABEND-AID "SNAP" dump, the following COBOL instruction should be used:

```
CALL 'SNAPAID'
```

During link edit of the program module concatenate ABEND-Aid's load library in the "SYSLIB" DD statement.

For simple compilations:

//SYSLIB DD DSNAME=SYS2.ABENDAID.LOADLIB,DISP=SHR

During execution of the program module you need to code the following DD statement:

//ABENDAID DD SYSOUT=*

Compuware Shared Services (CSS)

Compuware Shared Services is a set of components used by Abend-AID to provide storage, retrieval, and maintenance for abend reports, and source listings on datasets called DDIO Files. It consists of the following components:

1. DDIO Files
2. Compuware Viewing Facility (Compuware/VF)
3. Compuware Language Processor

DDIO Files

When allocating DDIO files, choose VSAM or sequential format; sample JCL for both are listed later in this chapter.

When compiling COBOL programs that are to be used with this product choose either a preprocessor or a postprocessor. Both have their advantages/disadvantage. Refer to the reference material listed below before selecting which type to use.

To have your ABEND-Aid report sent to the right DDIO, code the following statement in your JCL.

//ABNLTERM DD DSN=your.report.ddio,DISP=SHR

Compuware Viewing Facility (COMPUWARE/VF)

The Compuware/VF utility allows you to view your previously compiled source listings and your Abend-AID reports separately or together. To invoke the View Facility, key **=D.VF** at the ISPF primary options panel.

Once in the View Facility you will be required to enter the report data set & source data set(s) you wish to view. After pressing ENTER, the "OUTPUT SELECTION MENU" will be displayed, this screen allows for the viewing of a particular portion of the report.

Compuware Language Processor

The Compuware Language Processor can be run by either of the following two methods:

- As a preprocessor
- As a postprocessor

Both methods require changes to the compile JCL.

Cataloged Procedures

Procedure	Description
CSSFMTDS	Formats all types of data sets
CBLCSSPR	Cobol preprocessor
CBLCSSPO	Cobol postprocessor
OPCCSSPO	Cobol/Optimizer postprocessor

References

Abend-Aid documentation is available online in BookManager. For help accessing or using BookManager, see the BookManager chapter of the Handbook.

Chapter 9. DOCU/TEXT

Introduction

DOCU/TEXT is a software product designed for use in TSO/E (ISPF). It is used to document jobs and systems. Reports are produced using information that is collected from jobs, PROCs, system catalogs, VTOCS, and other sources. These DOCU/TEXT reports provide detailed information about jobs as well as overview information for system wide analysis.

DOCU/TEXT consists of modules that form an integrated system. The Analysis Documents Module consists of a set of reports providing:

- System Overview
- Job Overview
- I/O List
- Cross References
- JCL Flowcharts
- Indexes of Jobs, Procedures, Programs, and Files

Operations Documents Module delineates system and job resource requirements. The reports generated are used to supplement or replace:

- Manual Job Setup and Restart Forms
- Report Distribution Information
- Magnetic Tape Usage Requirements
- Job Scheduling and Resource Requirements

Extended Functions Module enhances the features of Analysis Documents or Operations Documents with:

- Online Viewing and Query of Documentation
- Smart Display

Accessing DOCU/TEXT

In TSO/E ISPF, execution can be in foreground or batch mode. The user can select jobs to be documented, JCL to be scanned, and reports to be produced through ISPF panels.

To access DOCU/TEXT, type **=D.E**.

At the NITC DOCU/TEXT Entry panel, you are prompted to enter a Project and Group (Group is optional). After pressing enter, up to 15 partitioned data sets (PDS) are created automatically and are prefixed with the Project and Group qualifiers that you supplied. The Create panel to store reports uses these libraries. If you leave Project and Group blank, Project will default to your logon id.

Use of an invalid high-level qualifier will result in an ACF2 violation and will require you to logoff TSO and then log back in.

Primary Menu Options

The following options are displayed at the DOCU/TEXT Primary Menu panel:

```
 0  JOB STMT     - Job Statement and TSO Output Parms
 1  OLI QUERY    - ONLINE INFO Query
 2  CREATE       - Analysis Documents
 3  CREATE       - Operations Documents
 4  CREATE       - All Documents
 5  DISPLAY      - Analysis Documents
 6  DISPLAY      - Operations Documents
 7  DISPLAY      - All Documents
 8  USER TEXT    - User Text Maintenance Facility
 9  SELECTION    - Selection Lists Maintenance Facility
10  SDF MENU     - Short Description Facility
 X  EXIT         - EXIT
```

Job Statement panel-

This option establishes output parameters for TSO/E hardcopy output when executing foreground and JCL for batch execution of DOCU/TEXT. Output parameters must be established via the Hardcopy Output Characteristics panel and the Job Statement For Batch Submission panel before creating reports through the Create panel. Set up your Job Statement as shown below:

```
===> //xxxxxxB  JOB (account information),'job info',
===> //         MSGCLASS=R,TIME=nn,PRTY=nn,CLASS=x,NOTIFY=xxxx
===> /*ROUTE PRINT rrrr
===> //PROCLIB DD DSN=project.DOCUTEXT.PARMLIB,DISP=SHR
===> //*
```

Replace x's with values that are appropriate for your agency. The Project and Group you entered at the NITC DOCU/TEXT Entry panel will determine the PROCLIB data set name. If both Project and Group were left blank, use the following format:

```
//PROCLIB DD DSN=logonid.DOCUTEXT.PARMLIB,DISP=SHR
```

If Project was selected and Group was left blank, use the following format:

//PROCLIB DD DSN=project.DOCUTEXT.PARMLIB,DISP=SHR

If both Project and Group were entered, use the following format:

//PROCLIB DD DSN=project.group.DOCUTEXT.PARMLIB,DISP=SHR

The PROCLIB statement is necessary to point DOCU/TEXT to your files; instead of the system default files. Failure to use the PROCLIB statement will result in a CA-ACF2 violation when trying to write reports to the files.

OLI Query panel-

Allows selective retrieval and display of critical information about batch jobs in a production environment. Information to be viewed must first be created through the Create panel.

Create panels-

Create any combination of the 15 DOCU/TEXT reports. The output can be directed or stored in disk files allocated by the Entry panel.

Display panels-

Allows you to view reports created and stored in PDSOUT members by the Create panel.

User Text panel-

Allows you to add helpful information to reports and online documents.

Selection panel-

Allows you to create a list of members that contain execution JCL statements. A PDS library must have already been allocated via ISPF before accessing this panel. The Selection option can only create list members in an existing library.

Short Description Facility (SDF) panel-

Allows you to assign a descriptive phrase to any JOB, PROC, PGM, FILE, or report. SDF entries are limited to 32 characters.

Exit panel-

Exit DOCU/TEXT and return to the ISPF primary options panel.

HELP Facility-

Access DOCU/TEXT Help by pressing PF1 at the particular panel you need help with. You may also position your cursor on a field and press PF1 to get help on the field in question.

Press **PF3** to back you out of the help panels.

DOCU/TEXT Input

DOCU/TEXT uses existing sources for the majority of its data. This eliminates the need to create a special master file whose only purpose is to reflect the operational environment.

DOCU/TEXT can receive input from the following sources:

- Production JCL
- ISPF Panels
- Cataloged Procedures
- Utility Control Library
- VTOCS
- System Catalogs
- Set-up
- Program Narratives
- Tape Control Information

DOCU/TEXT Output

DOCU/TEXT provides a wide variety of report documents to support both analysis and operation requirements. These reports can be stored in a PDS for later viewing or a hardcopy can be produced. DOCU/TEXT will allocate the PDS required for storage of these reports based on user's high-level qualifier.

Analysis Documents include:

- Table of Contents
- Index
- I/O Listing
- Data Set XREF
- Report XREF
- Program XREF
- Proc XREF
- Flowcharts
- Options in Effect
- Structured JCL Listing

Operations Documents include:

- Job Documentation
- Tape VOLSER Analysis
- Tape Pull List

- Report Analysis
- Options in Effect
- Structured JCL Listing

References

DOCU/TEXT manuals are available online in BookManager. For help accessing and using BookManager, see the BookManager chapter of the Handbook.

You may also purchase the following manuals from Diversified Software Systems, Inc.:

PD-12071-001 DOCU/TEXT Administration Guide
PD-12030-00-2 DOCU/TEXT User Guide

Diversified Software Systems, Inc.
18635 Sutter Boulevard
Morgan Hill, CA 95037

(408) 778-0896

Chapter 10. DCDIII

Introduction

Data Correlation and Documentation (DCDIII) is a maintenance tool for use in the ongoing maintenance of COBOL programs, DCDIII documents programs and groups of programs. It aids in design, development, debugging and maintenance through the use of Listing and Report Facilities.

DCDIII, at NITC, is available in the batch environment only.

Alternate Compile Listing Facility

1. The Alternate Compile Listing Facility provides an alternative to using the standard compile listing of the COBOL compiler and provides the following reports:

- Source listing (Replaces the compiler listing)
- CALL statements
- COPY statements
- Figurative Constants
- Literals
- Special Registers
- Data Division Condensed Cross Reference
  (Replaces compiler data name cross reference)
- Procedure Division Condensed Cross Reference
  (Replaces compiler procedure name cross reference)

2. The Other COBOL Reports Facility provides a method for reporting (on one or several programs together) the use of data throughout the program and system. The following COBOL reports are available through the Other COBOL Reports Facility:

- CALL Analysis Report
- COPY Analysis Report
- System Data Name Cross Reference
- System Cross Reference for Figurative Constants
- Table of Contents for Layouts
- Layout Report
- System Cross Reference for Literals
- System Paragraph Cross Reference
- 01 Record Report
- System Cross Reference Report for Special Registers

- System Record Analysis Report
- Verb Analysis Report

3. The JCL PROC Analysis Reports Facility provides a method for producing limited reports on existing JCL PROCS. There is only one basic JCL PROC Analysis Report.

Alternate Compile Listing Facility Cataloged Procedures

Procedure	Description
COMACL	Compile mode, input is compiler's output source listings
DCDACL	Independent mode w/ input from a partitioned data set
LIBACL	Independent mode w/ input from a librarian program library

Sample COMACL JCL

```
//job statement ….
//STEP1   EXEC COBUC,PARM.COB='XREF,LOAD,BUF=32000'
//COB.SYSPRINT DD DSN=&&COMPLIST,
//    DISP=(MOD,PASS),UNIT=SYSDA,
//    SPACE=(CYL,(10,2)),
//    DCB=(RECFM=FB,LRECL=121,BLKSIZE=1210)
//COB.SYSIN DD DSN=your.source.lib(member),DISP=SHR
//STEP2   EXEC COMACL
```

Sample DCDACL JCL

```
//job statement…
//STEP1   EXEC DCDACL
//DCD.COBOLIN DD DSN=your.source.lib,DISP=SHR
//DCD.COPYLIB DD DSN=your.copylib,DISP=SHR
```

Sample LIBACL JCL

```
//job statement ….
//STEP1   EXEC LIBACL,SRCLIB='your.librarian.file'
//LIB.SYSIN DD *
-SEL MEMBER,PSWD
-EMOD
-END
/*
```

Other COBOL Reports Facility Cataloged Procedures

Procedure	Description
DCDCOBOL	Multiple source program analysis w/ input from a partitioned data set.
LIBCOBOL	Multiple source program analysis w/ input from librarian program library.

MBRFETCH	Source program analysis member/library fetcher takes input from partitioned data set(s) and creates a sequential file that may be input to DCDCOBOL.

Sample DCDCOBOL JCL

```
//job statement …
//* If you use a copylib statement, make sure that all
//* COBOL members use the same copy lib file
//STEP1   EXEC DCDCOBOL,DATA=,COPY=
//DCD.COBOLIN DD DSN=your.source.lib1(member),DISP=SHR
//          DD DSN=your.source.lib2(member),DISP=SHR
//DCD.COPYLIB DD DSN=your.copylib,DISP=SHR
```

Sample LIBCOBOL JCL

```
//job statement …
//STEP1   EXEC LIBCOBOL,SRCLIB='your.librarian.file',
//          CALL=,COPY=,VERB=
//LIB.SYSIN DD *
-OPT EXEC
-SEL MEMBER,PSWD
-EMOD
-END
```

Sample MBRFETCH JCL

```
//job statement ….
//STEP1   EXEC MBRFETCH
//MBR.CLTCDMBR DD *
INDD=USERDD=COBOL=MEMBER
//MBR.USERDD DD DSN=YOUR.SOURCE.LIB,DISP=SHR
//STEP2   EXEC DCDCOBOL,RECORDS=,LAYOUT=
//DCD.COBOLIN DD DSN=&&PASSFILE,DISP=SHR
//DCD.COPYLIB DD DSN=YOUR.COPYLIB,DISP=SHR
```

JCL Proc Analysis Reports Analysis Cataloged Procedure

Procedure	Description
DCDJCL	JCL Procedures analysis

Sample DCDJCL JCL

```
//job statement …
//STEP1   EXEC DCDJCL
//MBR.CLTCDMBR DD *
INDD=USERDD=MEMBER
MEMBERNAME
//MBR.USERDD DD DSN=YOUR.JCL.LIB,DISP=SHR
```

References

To order DCDIII customer manuals, contact:

Marble Computer, Inc.
Willow Hill Professional Building
160-4 Dover Road
Chichester, NH 03234

Phone: 800-252-1400
or 603-798-4100

FAX: 603-798-5100

Please indicate on the purchase order that your agency uses DCDIII at the USDA, National Information Technology Center (NITC).

Chapter 11. MHTRAN-2

Introduction

MHtran-2 is a COBOL source code translator. MHtran-2 can aid in the translation of application programs from MVS COBOL. IBM COBOL, as used in this document, refers to IBM COBOL for MVS and VM and IBM COBOL for OS/390 and VM.

IBM no longer supports the older OS/VS COBOL language or the VS COBOL II language. For more information on this, see the Program Creation and Execution chapter of this Handbook. If upgrading your programs to the newer IBM COBOL, please see the IBM COBOL chapter of this Handbook.

When using MHtran-2, the vendor states that this product is not always a 100% solution to your COBOL migration. You will need to compile and test the output source code of the translator. Testing your converted programs will allow you to catch any new/old compiler problems. The possibility of incorrect source code or a limitation of the translator does exist.

Batch MHTRAN-2

Data set KCUSER.MHT2.JCL contains sample JCL for executing MHtran-2. Member COBTRAN is JCL used for batch execution.

MHtran-2 is parameter driven. When migrating to MVS COBOL, the Target-Dialect parameter will be coded to "TARGET-DIALECT=OS/390". Other parms within this JCL may also have to be modified.

The User's Manual for Mhtran-2 is needed to effectively utilize all options.

References

To order reference manuals or for product support, contact:

PRINCE Software, Inc.
3 Pearl Court
Allendale, NJ 07401

(201) 934-0022

E-Mail: tecspt@PRRINCEsoftware.com

Fax: *(201) 934-5007 (manuals)*
(201) 934-0220 (product support)

url: http://www.princesoftware.com/

To facilitate your request, identify yourself as an online customer of the USDA's National Information Technology Center (NITC), located in Kansas City, Missouri.

Chapter 12. AD/Cycle C/370 COMPILER

Introduction

The IBM SAA AD/Cycle C/370 Compiler is the IBM SAA C compiler for the VM and MVS environments. Programmers can use the C/370 compiler and libraries to create and compile programs that:

- Exploit the unique features of the mainframe environment
- Utilize the facilities of a wide variety of existing programs
- Can be easily modified in the future
- Have portable source code that can be recompiled for different operating environments

Compiler Features

- Optimization option
- Full reentrancy, allows for modification of external or static data
- Dynamic linking of C run-time libraries
- IBM supplied procedures for compiling, linking, and loading
- Extended Architecture capability, allowing you to exploit the full 31-bit address space
- Source debugging using inspect for C/370 and PL/I
- C source code listings, showing what code the compiler has generated from your program
- Diagnostic messages to aid in problem analysis

Supported File Types

- Variable-length, Fixed-length and Undefined-length Records
- Sequential Data Organizations
- VSAM Data Organizations
- Memory Files

For the supported file types, C/370 provides:

1. Transparent access to all file types
2. Opening and creating files by data set name or DDNAME
3. Full support for stream I/O

Support for Other IBM Products

- Database 2 (DB2)

- Structured Query Language/Data System (SQL/DS(*))
- Information Management System (IMS)
- Interactive System Productivity Facility (ISPF)
- Graphical Data Display Manager (GDDM)
- Virtual Sequential Access Methods (VSAM)
- Transaction Processing Facility (TPF)
- Customer Information Control System/Enterprise Systems Architecture (CICS/ESA)

Interlanguage Calls

- Assembler
- VS Fortran
- PL/I
- COBOL

Usage

The AD/Cycle C/370 Compiler may be invoked in the foreground or background. To access the compiler in the foreground, enter **=4.16** from the ISPF Primary Options panel. The Compiler may be invoked in the background (batch) using the following cataloged procedures:

Cataloged Procedures

Procedure	Description
EDCC	Compile
EDCCL	Compile and link edit
EDCCLG	Compile and link edit and execute
EDCCPLG	Compile, prelink, link edit and execute
EDCL	Link
EDCLG	Link and execute
EDCPL	Prelink and link edit

Sample Compile JCL -

```
//job statement …
//STEP01  EXEC EDCC,CPARM='SOURCE'
//COMPILE.SYSIN  DD DSNAME=your.source.lib(member),DISP=SHR
//COMPILE.SYSLIN DD
DSNAME=your.object.lib(member),DISP=OLD,UNIT=
```

Sample Link JCL -

```
//job statement …
//STEP01  EXEC EDCL,LPARM='LIST,MAP,XREF'
//LKED.SYSLIN  DD DSNAME=your.obj.lib(member),DISP=SHR
//LKED.SYSLMOD DD
DSNAME=your.load.lib(member),DISP=OLD,UNIT=
```

Sample Compile and Link JCL -

```
//job statement …
//STEP01  EXEC
EDCCL,CPARM='SOURCE',LPARM='LIST,MAP,XREF'
//COMPILE.SYSIN  DD DSNAME=your.source.lib(member),DISP=SHR
//LKED.SYSLMOD   DD
DSNAME=your.load.lib(member),DISP=OLD,UNIT=
```

Sample Execution JCL -

```
//job statement …
//JOBLIB  DD DSNAME=your.load.lib,DISP=SHR
//STEP01  EXEC member
//SYSPRINT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
```

Please consult the C/370 programming guide and the User's Guide for more detailed information on the compile procedures listed above. See the References section of this chapter for information on accessing these manuals online.

Debugger Capabilities

INSPECT for C/370 is a source-level debugging tool used to help test and debug programs written in C/370, PL/I, or a combination of the two. INSPECT can be used interactively or in batch mode and allows you to:

- Simultaneously view the program source/listing while controlling program execution.
- Start and stop a program as needed.
- Monitor variables as they change during execution.
- Test and debug with little advance planning.
- Establish temporary variables during your INSPECT session, using C or PL/I syntax.
- Access all library functions, as well as, stdin, stdout, stderr, errno, and __amrc.
- Optionally record all INSPECT interactions to an external log file.
- Fully control the testing level desired.
- Interactively invoke user and library functions, including calls to functions written in Assembler, COBOL, FORTRAN, and PL/I.
- Fully debug code in nested blocks.
- Debug reentrant programs.
- Receive online help when using INSPECT interactively.

Interactive Debugger Features

- User tailored multiple windows for program source and other output.
- Visual trace of program execution with current line/statement and breakpoints highlighted.
- Split screen capability.
- View a log window, program source, and a monitor window simultaneously.
- Color and configure the three windows to meet your needs.

- Use ISPF to split the screen into two logical screens.

Invoking INSPECT

You must compile your C/370 program using the TEST compiler option. You may use the CPARM parameter in your batch compile JCL or the preprocessor directive #pragma options in your source program to specify the TEST option.

Sample Compile JCL -

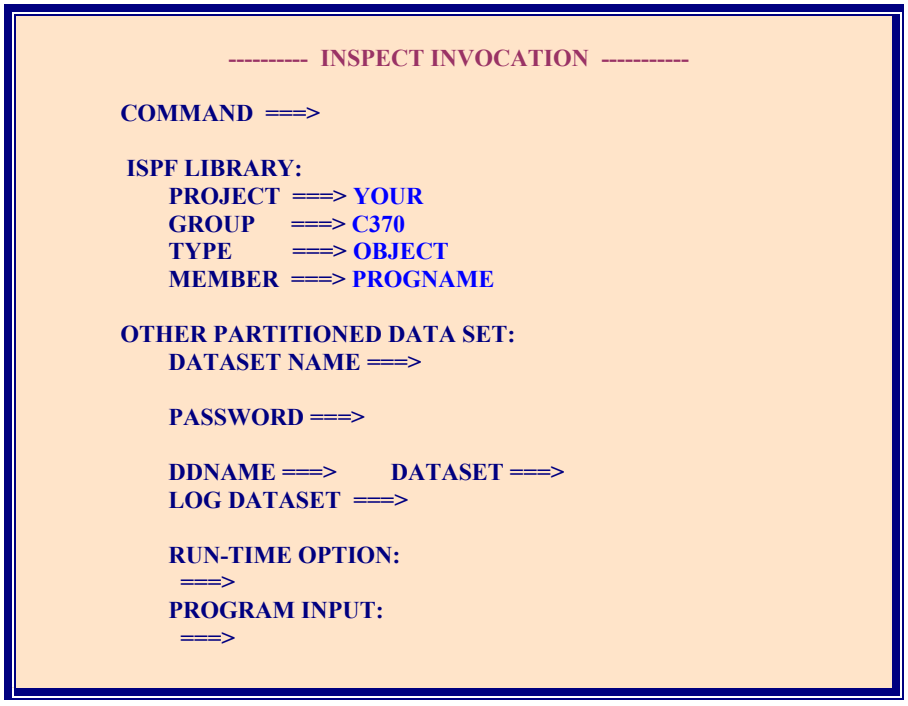
```
//job statement …
//STEP01  EXEC EDCC,CPARM='TEST'
//COMPILE.SYSIN  DD DSNAME=your.source.lib(member),DISP=SHR
//COMPILE.SYSLIN DD
DSNAME=your.object.lib(member),DISP=OLD,UNIT=
```

Sample C/370 Code -

```
#pragma options (TEST)
#pragma runopts (TEST(NONE,,;) EXECOPS)

#include <stdio.h>
#include <stdlib.h>
```

After successful compilation of your C/370 program with the TEST option, select option 4 from the ISPF/PDF Primary options panel. You will be in the FOREGROUND SELECTION panel. Select option IN (INSPECT). The following panel is displayed:



1. Enter the name of the object library containing your main program under ISPF LIBRARY or OTHER PARTITIONED DATA SET.

2. Enter a password only if your library is password protected.

3. Enter a ddname if you wish to have INSPECT allocate a data set for use with the DDNAME sub option of the TEST run-time option. DDNAME can be used as a reference for command files. It can also be used with the USE command. If you supply a DDNAME you must fill in the dataset field.

4. If a DDNAME has been supplied, you must enter the dataset name. This is the name of the COMMANDS data set to be allocated by INSPECT for use with the DDNAME sub option of the TEST run-time option.

5. Enter the name of your session log data set. If you leave it blank, it will be created with the name USER ID.INSPECT.LOG. This file contains a log of your INSPECT session.

6. Enter your runtime options. Runtime options are concatenated with the program input fields and passed to the PL/I OR C library at invocation. You can specify up to 70 characters. The run-time options and program input fields are limited to 140 characters combined.

7. Enter any input parameters for the main program. Both the run-time options and program input fields are concatenated and passed to the PL/I or C library at invocation. You can specify up to 100 characters. The run-time options and program input fields are limited to 140 characters combined.

8. After you have entered all necessary information, press the ENTER key. You have now invoked the INSPECT debugger.

9. To exit, enter QUIT at the command line and press the ENTER key.

References

Consult INSPECT and C/370 manuals for detailed information on compiler options, runtime options, and input parameters.

Online documentation for AD/Cycle C/370 is available using IBM's BookManager. For help accessing or using BookManager, see the BookManager chapter of this handbook.

Online training is also available for C Programming. See the Computer Based Training chapter of this Handbook for more information on accessing online training.

Chapter 13. CA-Intertest/Batch

Introduction

CA-InterTest/Batch is a full-screen, interactive, menu-driven testing and debugging facility for COBOL programmers. It complements Computer Associates' CA-InterTest/CICS product. CA-InterTest/Batch can be used to test COBOL programs compiled with the TEST option of the SAA AD/CYCLE COBOL/370 compiler.

Features

CA-InterTest/Batch grants the programmer complete access to the program as it is executing. The program can be stopped and the INTERCEPT panel displayed at user-specified intercept points. From the INTERCEPT panel, the user can browse the program listing in full-screen mode, using
program function keys (PFKs) to scroll backward or forward through the program. The programmer can examine and repair problem areas without ending the test session, or issue CA-InterTest/Batch commands to set new intercept points and resume testing. Each CA-InterTest/Batch panel is supported with a HELP panel. An online tutorial describes the CA-InterTest/Batch system.

- During a CA-InterTest/Batch test session, the programmer can:
- Stop the test at any executable PROCEDURE DIVISION statement in the program.
- View the program listing while the test session is stopped.
- Trace the program's execution and browse the trace table entries.
- Display and alter data-item values.
- Step through the program in increments of any number of statements.
- Check the number of times each statement in the program is executed.
- Step through the program execution in reverse order.
- Access the other CA-InterTest/Batch options, such as, Core.

InterTest Batch grants the programmer complete access to the program as it is executing. The program can be stopped and the INTERCEPT panel displayed at user-specified intercept points. From the INTERCEPT panel, the user can browse the program listing in full-screen mode, using program function keys (PFKs) to scroll backward or forward through the program.

The programmer can examine and repair problem areas without ending the test session, or issue InterTest/Batch commands to set new intercept points and resume testing.

During an InterTest/Batch test session, the programmer can:

1. Stop the test at any executable PROCEDURE DIVISION statement in the program.
2. View the program listing while the test session is stopped.
3. Trace the program's execution and browse the trace table entries.
4. Display and alter data-item values.
5. Step through the program in increments of any number of statements.
6. Check the number of times each statement in the program is executed.
7. Access the other InterTest/Batch options such as Core.

Invoking CA-Intertest/Batch

Access CA-Intertest/Batch by keying **=D.IB** from the ISPF Primary options panel.

Each InterTest/Batch panel is supported with a HELP panel. Typing HELP at the InterTest/Batch Primary panel will invoke an online tutorial describing the InterTest/Batch system.

References

Online documentation for CA-Intertest/Batch is available using IBM's BookManager. For help accessing or using BookManager, see the BookManager chapter of this handbook.

Manuals may also be ordered from Computer Associates by sending purchase orders to the following:

Computer Associates International
2291 Woodoak Drive
Herndon, VA 20171

Attn: Federal Division

(703) 709-4500

Please indicate in that you use the CA-Intertest/Batch installed at USDA National Information Technology Center in Kansas City.

Addition reference information may also be obtained by accessing url: http://www.ca.com/ and entering in the SEARCH box: **Intertest/Batch**.

Chapter 14. CA-Intertest/CICS

Introduction

CA-InterTest/CICS is a comprehensive CICS testing package that consists of five individual transactions:

- CNTL - used to control InterTest your CICS system.
- CORE - used to display, modify, and correct problems in areas of main storage.
- FILE - used to display, modify, and correct problems in areas of auxiliary storage.
- LIST - used to display the source listing and compiler output of a program.
- HELP - used to request assistance and answer questions concerning any of InterTest's commands of facilities while InterTest is running.

Functions of Intertest

1. Provides interactive debugging.
2. Stops the execution of a tested program.
3. Eliminates the need for addresses. Uses the symbolic name you wrote in the COBOL, PL/1, or Assembler language program.
4. Eliminates the need for dumps. There are no online dump requests or trace requests in your program.
5. Protects the CICS system. InterTest only allows the program to change the contents of a storage area that legitimately belongs to the tested transaction.
6. You can turn InterTest on at any time. It resides in Virtual Storage.
7. InterTest serves as an intermediary between the program you are executing and your CICS system.

CA-SYMDUMP

CA-SymDump (CA-InterTest's optional symbolic dump facility) is a powerful online tool that brings application dumps back to life. CA-SymDump works with CA-InterTest to improve your ability to analyze and resolve application program problems. With CA-SymDump, you can:

1. Analyze dumps symbolically using familiar CA-Intertest source code displays to resolve your application dumps.
2. Immediately resolve CICS production problems from any region, for example, debug production abends from your test region.
3. Manage your dump data set so you can selectively retain, view and print the dumps you need, and discard the ones you don't.

References

Manuals may also be ordered from Computer Associates by sending purchase orders to the following:

Computer Associates International
2291 Woodoak Drive
Herndon, VA 20171

Attn: Federal Division

(703) 709-4500

Please indicate in that you use the CA-Intertest/Batch installed at USDA National Information Technology Center in Kansas City.

Chapter 15. SAS/C COMPILER

Introduction

The SAS/C Compiler provides a C development environment for the IBM mainframe. The primary elements of the SAS/C Compiler are the compiler and the run-time library. However, the compiler product also includes a number of utility programs, as well as several configurations of the run-time library for specialized environments.

Although the compiler is heavily oriented to use in large software systems, it is an efficient tool for any software project that is written in the C language.

Features

- Generation of reentrant code, enabling many users to share the same code.
- An optimization phase to increase speed and efficiency of generated code.
- Generated code is executable in both 24-bit and 31-bit addressing modes, allowing compiled programs to execute above the 16-megabyte line.
- Generated code is identical for MVS and CMS operating systems, allowing compatibility without recompiling.
- Built-in functions, including many of the traditional string-handling functions, that generate in-line machine code rather than function calls.
- Support for low-level systems programming through in-line machine code and the Systems Programming Environment (SPE).
- Full support for inter-language communication.
- Dynamic loading of I/O; Modules are loaded as needed at runtime.
- Full-function debugger.

User Requirements

Knowledge of C language is required.

Batch SAS/C Compiler and Debugger

Cataloged Procedures

Procedure	Description
LC370CL	Compile and link edit
LC370CLG	Compile, link edit and execute
LC370D	Debug C Programs
LC370L	Link
LC370LG	Link and execute

LC370LR	CLink Linkage Editor Preprocessor and link edit
LC370LRG	CLink Linkage Editor Preprocessor, link edit and execute

Sample Compile JCL -

```
//job statement …
//STEP01  EXEC LC370C
//C.SYSIN   DD DSNAME=your.source.lib(member),DISP=SHR
//C.SYSLIN  DD DSNAME=your.object.lib(member),DISP=OLD
```

Sample Compile and Link JCL -

```
//job statement …
//STEP01  EXEC LC370CL
//C.SYSIN   DD DSNAME=your.source.lib(member),DISP=SHR
//LKED.SYSLMOD  DD DSNAME=your.load.lib(member),DISP=OLD
// UNIT=
```

Sample Compile, Link and Execute JCL -

```
//job statement …
//STEP01  EXEC LC370CLG
//C.SYSIN   DD DSNAME=your.source.lib(member),DISP=SHR
//LKED.SYSLMOD  DD DSNAME=your.object.lib(member),DISP=OLD
// UNIT=
```

Sample Compile, Link and Execute JCL (with Debug option) -

```
//job statement …
//STEP01  EXEC LC370CLG,PARM.C='DEBUG'
//C.SYSIN   DD DSNAME=your.source.lib(member),DISP=SHR
//LKED.SYSLMOD  DD DSNAME=your.object.lib(member),DISP=OLD
// UNIT=
//C.SYSDBLIB  DD DSNAME=your.sas.DBGLIB,DISP=OLD
```

Sample Execution JCL -

```
//job statement …
//JOBLIB  DD DSNAME=your.load.lib,DISP=SHR
//          DD DSNAME=SYS90R.SASC.V700C.LINKLIB,DISP=SHR
//STEP01  EXEC MEMBER
//SYSPRINT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
```

Consult the SAS/C Compiler and Library User's Guide for more information on the procedures listed above.

Training

The NITC offers online training courses on C Programming that may be helpful. Information on available Computer Based Training (CBT) and how to register may be obtained by browsing Teleview NEWS option 2 – Computer Based Training.

References

To order SAS/C manuals, contact:

SAS Institute, Inc.
SAS Campus Drive
Cary, N.C. 27513-8000

(919) 677-8000

Chapter 1. CA-7

Introduction

CA-7 is a powerful tool used for managing production control. The system automates management and processing, significantly improving Center productivity and customer services. As such, it has the capability to address a broad range of activities some of which are described in this chapter.

Facilities and Features

1. **Workload Scheduling -** The system can define and schedule all activities associated with production.

2. **Workload Sequencing and Control** - The system facility that prevents CPU jobs from being executed before necessary prior tasks are completed and all job dependencies are satisfied.

   Although the system automatically schedules and invokes the work defined to it, it is sometimes necessary to circumvent scheduled workflow for higher priority items. CA-7 permits the customer to handle unscheduled interruptions online so that the most current priorities can be rapidly addressed and resolved.

   Schedules can be moved forward or backward. Jobs and/or activities can be held up, be given a higher priority, or be cancelled online. This is accomplished without time consuming rescheduling.

   Online prompting reminds the appropriate customer(s) when scheduled work is late, or in danger of becoming late. Prompting promotes a smooth flow of work by drawing attention to required action and/or activities that might otherwise be overlooked or delayed.

   The JCL management facility can include or omit override statements, depending upon job requirements. Both scheduled and unscheduled override requirements are supported within the system. A JCL syntax check facility is provided to test the accuracy of the JCL prior to formal processing.

3. **Job Restart** - Jobs that would usually terminate in an abnormal manner are moved to the system's request queue for job restarts. All such jobs may be listed online together with the necessary job restart information.

   Job restart information identifies the last step successfully executed, the abend code and restart steps. After necessary corrections, abnormally terminated jobs can be restarted online.

Within the system, only one transaction is necessary to restart. This can be accomplished online.

4. **Editor** - Editor is an interactive system facility for creating, modifying, and managing 80 character image data. With Editor, JCL can be created and submitted for processing online. Editor can also be used to write customer exists and to add documentation to the system database.

5. **Online Utility Execution** – The CA-7 utilities allow online execution of commonly used IEHPROGM-type functions. The system's utilities allow online execution of commonly used IEHPROGM-type functions. The most common are:

   - Data Set Allocation
   - Allocate and Catalog
   - BLDG
   - Catalog
   - Rename
   - Scratch
   - Uncatalog
   - List PDS
   - List Catalog

   Online executions of these utilities reduce the time that would otherwise be required to perform these functions.

6. **Security** - CA-7 supports both internal and external security. External security packages, such as CA-ACF2 and CA-Top Secret, are used to control access to CA-7, which terminals an operator can use, which command an operator can issue, which data sets an operator can access, and the authority associated to jobs submitted by CA-7.

   The internal security provides five levels of security. They are:

   - Terminal/Operator
   - Operator/Application
   - Application/Command
   - Command/Function
   - User ID/External Data Set

   Through the security subsystem, customer personnel are allowed access only to those functions of the system related to their job(s).

7. **Workload Forecasting** - The Workload Forecast facility provides several important functions. It allows you to:

   - Project the scheduled workload for a given period of time.
   - Verify newly defined schedules.

- Produce a hardcopy checklist of scheduled tasks in a worksheet format.

8. **History Reporting -** As work is processed under CA-7, activities are recorded in a Log data set. This information may be used to generate a variety of reports through the CA-7 History Reporting facility.

9. **Workload Documentation** - CA-7 allows documentation of the workload at many levels, from general descriptions to specific instructions. Documentation is added to the CA-7 workload definition through the CA-7 text editor and is available for reference at CA-7 terminals. Including documentation in a system database is optional.

10. **Management Level Reporting** - The system collects job statistics on significant events throughout the system. Customers are allowed to define additional events requiring tracking. A report facility exits to provide graphs on the data collected according to customer or system specifications if desired. Such reports are available either on-line for immediate review or in batch mode for hard copy reports.

11. **Virtual Resource Management (VRM)** - provides flexible job submission control and is in effect only for those jobs that have been defined to CA-7 as using VRM resources. VRM resources can be freed at job or step termination thus allowing for better resource control. VRM features multiple resource types and multiple resource dispositions. To access VRM panels use the RM Option on the CA-7 MENU.

    Since many agencies can use this feature it is recommended that you prefix or append your agency abbreviation to the resource name. EX.(agabb_test1).

    For more information on VRM please see the chapter in the CA-7 Database Maintenance Manual.

12. **Automated recovery Facility (ARF)** – used to monitor production jobs for exception conditions and schedule recovery actions at or near point of failure. To access ARF panels use the AR.3 Option on the CA-7 MENU.

    Types of exceptions the ARF can monitor are:

    a. Abends at the job or step level
    b. Job elapsed time
    c. Jobs with a CA-7 late status

    Types of recovery that can be performed are:

    a. Scheduling and tracking recovery jobs
    b. Issue message to the TSO user or the MVS console
    c. Issue CA-7 commands
    d. Set the final disposition of the job as restart, cancel, or Force complete.

For more information on ARF please see the chapter in the CA-7 Database Maintenance Manual

13. **CA-DRIVER** - This feature provides extensive JCL tailoring and control statement setup capabilities. Using CA-Driver can greatly reduce or eliminate the need to make JCL modifications prior to job scheduling and submission. CA-Driver features reserved-name variables that can be referenced in a CA-Driver procedure. For information on how to use CA-Driver see the chapter in the CA-7 Database Maintenance Guide.

CA-7 Access

CA-7 may be accessed through ISPF panels by entering **=C7** from the ISPF Primary Options panel.

CA-7 may also be accessed via application id (applid) UCC07. This application may be added to your Teleview profile (menu). If you wish to have CA-7 added to your Teleview profile, contact your agency IT security staff to request it.

Online Help/Tutorial

Online Help is available in CA-7. Typing HELP from the CA-7 Primary Options Menu panel will provide general information on CA-7/TSO-ISPF.

Typing HELP from the Online CA-7 panel (option 1) provides a CA-7 Online Tutorial.

Change Group ID Access

CA-7 allows users the capability to use multiple UID groups. A user may switch to a different group they have access to (other than their default group) by specifying a UID resource at the time of accessing CA-7.

This is accomplished by placing the UID resource in the form of CA70xxx (xxx is the UID resource number) in the UID resource field on the CA-7 Primary Options Menu before entering the Online facility.

Function Keys

Program function (PF) keys may be assigned to define a specific command for a predetermined function. This enables the terminal user to enter common functions with a single keystroke per function.

To make changes to your PF Keys, select **0** from the CA-7 Primary Option Menu panel.

Training

Online training is available for CA-7. See Teleview News, Option 2 – Computer Based Training, for more information regarding online training or contact the NITC Customer Support Center at 816-926-6681 for assistance.

References

CA-7 manuals may viewed from BookManager. For information on how to access and use BookManager, see the BookManager chapter of this handbook.

Additional reference information may also be obtained by accessing url: http://www.ca.com/ and entering in the SEARCH box: **ca-7**.

The following reference manuals may also be ordered directly from Computer Associates:

CA-7 Commands Guide
CA-7 Database Maintenance Guide
CA-7 Messages Guide
CA-7 Systems Programmer Guide
CA-7 Reports Guide
CA-7 Primer

To order the manuals, write or call:

Computer Associates International
2291 Woodoak Drive
Herndon, VA 20171
Attn: Federal Division

Phone: (800) 225-5224

Please indicate that your agency uses CA-7 that is installed at the National Information Technology Center to allow for direct ordering and billing.

Chapter 2. CA-11

Introduction

The major function of the CA-11 Online System is to display information relative to shop-wide production status and individual job status. Additionally, a batch interface is provided. This permits all the Online System functions to be initiated in batch mode. All system responses could then be directed to an output printer. This is useful to obtain hardcopy of valuable output, such as, production status at shift changes.

Usage

The proper management of rerun operations is a very important part of the overall thrust of production control and optimal resource utilization. CA-11, the automated rerun and tracking system (ARTS), provides for the efficient management of reruns in the data processing environment. This management tool provides monitoring, reporting, and control of necessary reruns. CA-11 supports two (2) basic functions: rerun tracking and rerun handling.

Interactive Access

CA-11 may be accessed from the ISPF Primary Options panel by selecting **=D.11**.

From the CA-11 menu:

Online Help
Online Arts Commands
Restart Screen

Restart Screen

This screen is similar in function to the restart (xrst) screen in CA-7. It will provide and ISPF/PDF restart support facility. Access to restart information within CA-11 is provided. And will allow a user to reset parameters as needed.

RMM Interface

CA-11 will expire tapes that will be recreated by a rerun/restart processing when RMM is to be used.

Rerun Tracking

Rerun tracking permits the customer to quantify the impact that reruns will have on production, to determine what the causes of reruns are, and to identify recurring causes. With this type of information in hand, the customer can institute a systematic approach to reducing the number of necessary reruns.

Rerun Handling

Rerun handling permits the customer to free personnel from the manual procedures normally used to setup and process reruns. CA-11 automatically handles all catalog maintenance, data set maintenance, and other necessary rerun adjustments. Step restarts and complete job reruns can
be handled with equal ease.

Auto Setup

Auto setup is an optional system feature used to automatically determine where abended jobs should be restarted. With this option, a customer can resubmit a job without any JCL changes. The system will then restart the job at the step. Should a step be non-restartable, the restart will commence at the most recently executed restartable step prior to the abended step.

Auto setup is the installed default. All jobs will be processed as installed unless the customer explicitly indicates that a particular job is not to assume the default. To do this, the customer must designate the auto setup action for the particular job. The auto setup status for an individual job may be changed by setting flags in the job's control record in the job execution history file (JEHF).

Job Status

CA-11 maintains the status of all jobs being tracked by the system. The status is defined to be one of four states:

Job Status	Description
A	Unresolved abend in last run. Run terminated with a user or system abend and has not been set for rerun.
B	Completed run. Run did not abend.
C	Executing.
S	Set-for-rerun. The job is set to be rerun.

Reason-for-Rerun

The reason-for-rerun field is a required option. It is required to produce meaningful management reports. Reason-for-rerun is one of six fields that may be selected for sorting when producing batch reports.

When used as a sort field, the various reasons or causes for job failure, and the number of each, can quickly be determined. The reason field is 40 bytes long and can accommodate text and customer-defined codes or other appropriate tracking means.

Reason-for-rerun codes and their corresponding text can be found in system table UCC11REA. Customer production control personnel should become familiar with the appropriate codes for rerun entry.

When no rerun/restart reason is supplied a job will abend with the following message:

U11-025 CA-11 - Reason for rerun for job XXXXX not provided.

RMS Batch Step

To invoke CA-11 without using CA-7 you will need to code one of the following as the first step of your job.

For a production run:

```
//CA11  EXEC UCC11,TYPRUN='P'
```

For a Restart or Rerun:

```
//CA11  EXEC UCC11,TYPRUN='P,stepx,RE=/reason/'
```

There are many other parameters you can supply based on your requirements.

Training

The CA-11 product provides an online tutorial. Type **T** at the CA-11 Selection Menu to invoke the tutorial.

Online training is available for CA-7. See Teleview News, Option 2 – Computer Based Training, for more information regarding online training or contact the NITC Customer Support Center at 816-926-6681 for assistance.

References

CA-11 manuals may viewed from BookManager. For information on how to access and use BookManager, see the BookManager chapter of this handbook.

Additional reference information may also be obtained by accessing url: http://www.ca.com/ and entering in the SEARCH box: **ca-11**.

The following reference manuals may also be ordered directly from Computer Associates:

CA-11 Message Guide
CA-11 User Guide
CA-11 System Programmer Guide

To order the manuals, write or call:

Computer Associates International
2291 Woodoak Drive
Herndon, VA 20171
Attn: Federal Division

Phone: (800) 225-5224

Please indicate that your agency uses CA-11, installed at the National Information Technology Center, to allow for direct ordering and billing.

Chapter 3. CA-7/CA-11 Interface

Introduction

CA-7 has a job restart facility used to restart jobs that failed during execution. It also can be used to rerun jobs that processed incorrectly. The restart facility is also an interface to CA-11 (ARTS).

Restart List

A restart list can be generated which contains the following information:

- Abend code or JCL error indicator.
- Identification of the last successfully executed step.
- A list of restartable steps. When determining step restartability CA-7 considers passing temporary data sets and backward references, condition coding and update in place of direct access files that may make steps non-restartable.
- Messages giving information about restartability.
- Customers supplied job documentation identified as restart procedure documentation.

Restart Procedures

When a job fails, CA-7 sets a restart requirement to prevent the job from being resubmitted. It moves the job back to the request queue. Methods by which a job may be restarted are as follows:

1. Non CA-11 restart. If CA-11 is not available, the appropriate customer personnel should under take recovery procedures. When a job is ready to restart, the CA-7 restart function is invoked. Using this formatted screen, the customer can tell CA-7 the restart step and or proc. After the restart, information and reason for restart is supplied, CA-7 will resubmit the job.

2. CA-11 restart. To invoke the CA-11 interface you will need to modify the job screen to reflect "INSERT-RMS" to "Y". When CA-11 is used for restart, CA-7 provides a formatted screen on which CA-11 parameter data is supplied. Once restart information is verified and/or modified CA-7 resubmits the job.

3. ARTS online inquiry is also provided.

References

CA-7 Release 3.2
CA-7 Commands Guide
CA-7 Database Maintenance Guide
CA-7 Messages Guide
CA-7 Systems Programmers Guide
CA-7 Reports Guide
CA-7 Interfaces Guide
CA-7 Primer
CA-11 Release 2.1
CA-11 Message Guide
CA-11 User Guide
CA-11 System Programmers Guide

To order manuals, write or call:

Computer Associates International
2291 Woodoak Drive
Herndon, VA 20171
Attn: Federal Division

Phone: (800) 225-5224

Please indicate that your agency uses CA-11, installed at the National Information Technology Center, to allow for direct ordering and billing.

Chapter 1. ORACLE RDBMS

Introduction

ORACLE RDMS is a relational database management system (RDBMS) based on Structured Query Language (SQL) and is capable of handling online transaction processing in a distributed environment. ORACLE is written in the C language and runs on a wide range of mainframes, minicomputers and microcomputers and on over many different operating systems. Within this heterogeneous environment, ORACLE applications can move from one machine to another with virtually no modification.

Features

ORACLE RDBMS provides the following features:

1. Table-Like Representation of data
2. 4th Generation Language
3. Full Relational Capabilities
   a. Automatic Navigation (optimized)
   b. All Relational Operators
4. Flexibility
   a. Easily modify data
   b. Easily change of database structure
5. Integrated Data Dictionary
6. Compatibility
   a. ANSI standards
   b. FIPS PUB 127
   c. ANSI SQL
7. Portability
   a. Micro
   b. Mini
   c. Mainframe
8. Connectivity
   a. Distributed processing
   b. Distributed database capability

ORACLE Product Descriptions

SQL*PLUS is Oracle's interactive interface for ad hoc queries and database creation and maintenance. SQL*PLUS accepts all SQL commands and provides additional commands that support report formatting, hard copy output, execution of stored queries, and modification of current queries.

SQL*FORMS is a tool for quickly developing forms-based applications for entering, querying, updating and deleting data. Application needs are specified, by using simple menus and a screen painter, instead of writing programs. SQL*FORMS combines your instructions with information from the data dictionary to generate your application. SQL*FORMS uses a non-procedural approach that allows you to prototype applications effectively. You have the flexibility to quickly adjust your SQL*FORMS application to changing requirements. See the SQL*FORMS section of this chapter for more information on SQL*FORMS. Note: SQL*FORMS is no longer available for OS/390 platform with the release of Oracle 8.

PRO*C Database access through procedural languages is provided by Oracle's precompilers and procedural interfaces. ORACLE precompilers allow users to access and manipulate data in an ORACLE database using familiar programming languages. Users embed SQL statements in their C/C++ programs and the precompiler translates the statements into the appropriate C/C++ source code.

PRO*COBOL Database access through procedural languages is provided by Oracle's precompilers and procedural interfaces. ORACLE precompilers allow users to access and manipulate data in an ORACLE database using familiar programming languages. Users embed SQL statements in their COBOL programs and the precompiler translates the statements into the appropriate COBOL source code.

SQL*LOADER facilitates the transfer of data from OS.390 datasets into tables in an Oracle database.

SQL*NET handles all communications for ORACLE between various types of machines. It is used to distribute processing by connecting applications software on one computer to that of another computer. Protocols ASYNC and 3270 have been installed.

CICS ATTACH is a multi-threaded facility that allows CICS transactions to access an ORACLE for MVS database. CICS Attach is implemented as a CICS task related user exit.

Customers who would like to use the CICS Attach facility for their ORACLE application must notify the Information Management and Customer Support Division at (816) 926-6681 and provide ample time to plan application requirements for the CICS region.

ORACLE ACCESS MANAGER allows applications running under CICS to issue industry-standard SQL statements against an Oracle server, without coding proprietary application program interfaces (APIs).

ORACLE CALL INTERFACE is a set of APIs allowing manipulation of data and schema in an Oracle database. OCI is compatible with:

IBM SAA AD/Cycle and IBM COBOL
IBM C/370, IBM C/C++, and SAS/C
IBM System 370 Assembler

ORACLE REPORTS allows previously defined reports in batch under OS/390. The reports must have been developed on a non-OS/390 platform supporting its definition component. Oracle Reports replaces SQL*ReportWriter for OS/390.

SQL*PLUS enables manipulation of data using SQL commands. SQL allows for the following:

Enter, edit, store retrieve, and run SQL commands
Format, perform calculations on, store, and print query results in report form
List column definitions for any table
Access and copy data between databases
Send messages to and accept responses form a user

SQL*REPORT produces reports more powerful than SQL*Plus reports.

Policy

ORACLE is available for all customer agencies of the Center. The agency is requested to specify an ORACLE administrator to serve as the designated point of contact with the NITC for ORACLE implementation and related issues.

The Center will establish and maintain a test and production instance of ORACLE for use by customer agencies. The Agency Oracle Administrator must provide the Center with the following application specifications to ensure proper resources are allocated prior to beginning development or placing the application within the production environment:

1. High-level qualifier (HQL) for the table space(s) the agency will be using and associated accounting information. The Center must have CA-ACF2 access to the files associated with the agency table space(s).

2. Number of tables, number of rows and length, and expected growth of each table.

3. Number of developers, and end users.

The NITC will work with the agency administrator to determine whether or not the agency's ORACLE workload would require the establishment of an additional ORACLE instance.

The Center will establish all ORACLE instances and assign the associated ORACLE subsystem name (SSN). The subsystem name is a maximum of four (4) characters. The first two positions will be the agency identifier, the third position will be "D" for development, "T" for test or "P" for production, and the fourth position will be a numeric identifier. All agencies requiring a separate instance of ORACLE must provide NITC with access to a high level qualifier, file sizes for the dictionary, control file, and log files.

Each ORACLE instance is associated with an authorized load library. The Center will own and maintain all authorized load libraries. Authorization will not be granted to any user owned library.

Naming Conventions

A standard naming convention must be maintained for all datasets associated with an ORACLE instance. The naming convention allows DASD management to easily identify ORACLE data sets for optimum performance and emergency notification in case of media failure. The following data set naming convention must be used:

HLQ.SSN.xxxxxx	HLQ	is the High-level qualifier
	SSN	is a 4-position subsystem name assigned by NITC associated with the ORACLE instance

Authorization and Security

Oracle may be accessed via the user's TSO/E logon id and may be converted to an OPS$ signon within Oracle (sample OPS$xxxxxx; where xxxxxx is a TSO/E logon id). Oracle can also be accessed from a client to the mainframe requiring an Oracle username. All Oracle usernames have passwords, which may or may not be the same as the TSO/E password.

Agencies requiring access to ORACLE should send a request or in writing or email to the Chief, Information Management and Customer Support Division. The request should contain the name of the person(s) needing access, their TSO/E logonid (if applicable), account code, agency name, which database is to be accessed, and which of the following ORACLE permissions are required.

- **RESOURCE** - Allows the user to create database objects, such as tables, indexes, clusters and sequences. User can also grant to, or revoke from, other users' privileges on his/her objects.

- **CONNECT -** Allows the user to query other users' tables, if select access has been granted. The user can also perform data manipulation (INSERT, UPDATE, DELETE) on other users' tables, if the appropriate access has been granted. Users with CONNECT privilege may create any tables, clusters, sequences or indexes within the users schema.

- **OTHER PERMISSIONS** such as web toolkit packages, special roles or grants needed to perform job functions

The Center will maintain the DBA functions for the test and production instance of ORACLE. Agencies with a separate ORACLE instance are responsible for the DBA functions within their instance unless special arrangements have been made for NITC to provide DBA services.

DBA functions include database creation, startup, shutdown, recovery, backup, file and/or directories cleanup, enrollment of new users and system monitoring.

The NITC advises that all agencies consider the use of a generic user ID(s) for ORACLE administrators. The internal security of ORACLE is tied to the user ID that is granting

and revoking privileges. Revocation of privileges granted to an administrator will cause the automatic revocation of all privileges granted by that administrator.

Job Classes

ORACLE runs in the MVS/ESA environment. An instance can be run as a started task or as a batch job in job class 'M'. ORACLE redo log archive jobs must be submitted to run in class 'J'. Class 'M' is restricted to batch jobs for ORACLE instances. Class 'J' jobs are restricted to archive jobs.

Access

Customers may access test or production instances by selecting ISPF option, **=P.OR**. The ORACLE panel requires entry of the ORACLE product desired and the subsystem name associated with the ORACLE instance you are authorized to use.

An incorrect product code or subsystem name will result in an error requesting the reentering of data. If you are unsure of the proper subsystem name, contact your agency ORACLE administrator.

Some common errors received when logging on to ORACLE are:

1. ORA-01017 Invalid username/password; logon denied
   Cause - Invalid username and/or password
   Action – Verify user ID and password. Enter a valid username/password.

2. ORA-01033 Oracle initialization or shutdown in progress
   Cause - Oracle is being started or shutdown
   Action - Wait a few minutes and retry

3. ORA-01035 Oracle only available to user with DBA privileges
   Cause - Oracle has been started in DBA mode
   Action - Contact the NITC or your ORACLE administrator

4. ORA-4115 Oracle subsystem not active
   Cause - Specified Oracle subsystem is not active
   Action - Contact the NITC or your ORACLE administrator

5. ORA-4119 Maximum user count exceeded
   Cause - Maximum number of users are signed on instance
   Action - Wait a few minutes and retry

6. ORA-12154 TNS: Could not resolve service name
   Cause - The service name specified is not in the TNSNAMES.ORA on the client computer.
   Action – Verify that the TNSNAMES.ORA file exists in the proper location, verify the service is listed in the TNSNAMES.ORA file, verify there are no syntax errors in the file.

7. ORA-12203 TNS: Unable to connect to destination
   Cause - Invalid TNS address provided or destination is not listening
   Action – Verify that the correct service name was entered; contact the database administrator for listener availability.

Cataloged Procedures

Procedure	*Description*
OPCCCOB	Oracle COBOL PreEdit
OPCCCOBL	Oracle COBOL PreEdit, Compile, Link
ORALDR	SQL*Loader Batch

The Center will create public procedures to assist ORACLE users in the operation and maintenance of their instance. These procedures will be developed on an as needed basis.

ORACLE REPORTS

Oracle Reports (formerly SQL*ReportWriter) for MVS is an ORACLE tool that accepts a report definition program developed in another environment and allows it to be run against an ORACLE database on MVS. It has no development facility of its own on the MVS platform.

There are five steps required to load and run reports on MVS:

1. An Oracle Reports program must be developed under PC, VAX, or any other environment that supports Oracle Reports. The same table structure must exist on both platforms of the Oracle Reports facility.

2. A report export file (REX file) must be created via DUMPREP facility. The export file must be transferred to a sequential or partition data set on the MVS platform. The file transfer program must support text and perform ASCII to EBCDIC conversion.

3. The converted file is then loaded into the centralized report writer tables on MVS via LOADREP facility. The ORACLE logonid is flagged as the owner of the reports loaded.

4. The compiled report structure file (REP file) must be generated from the report definitions stored in the centralized report writer tables using GENREP facility.

5. An output report is then created from the compiled report structure file (REP file) via RUNREP facility.

ORACLE REPORTS Components

The following procs have been created to unload, load, generate and run reports and placed into the system proclib. An additional PROC (GENRUN) used for executing GENREP and RUNREP as one job has been included.

- DUMPREP unloads a report definition into an ASCII export file for transporting to an ORACLE database.
- GENREP generates an executable report file.
- LOADREP loads an exported report definition into an ORACLE database
- MOVEREP converts Oracle Reports from Version 1.0 to Version 1.1.
- PRINTDEF crates printer definitions
- RUNREP executes previously defined reports

Reports may be run via a CLIST and sent to the screen for verification and then deleted or sent to a printer. A generic CLIST for all NITC customers is not possible due to the report parameters needed to be passed.

The Center can provide you with a model CLIST that can be modified to conform to the requirements of your reports. For additional information, contact the Information Management and Customer Support Division at (816) 926-6681.

ORACLE SQL*FORMS

SQL*Forms for MVS is a set of ORACLE tools to create, maintain, and invoke SQL*Forms applications. Note. SQL*FORMS is not compatible with MVS beyond Oracle release 7.3.3 (unsupported as of December 31, 1999).

SQL*Forms Cataloged Procedures

Procedure	Description
ORACVT2	Convert form to CICS load module
ORAIAC30	Convert V2.3(INP) file to V3.0(INP)
ORAIAG30	Generates an executable V3.0(FRM) from an (INP) file
RSCCNV	Create load modules for ORACLE*Terminal files

SQL*Forms Interactive Access

SQL*Forms may be built using the interactive forms design tool found on the ISPF ORACLE DEVELOPMENT TOOLS menu panel.

Select **=P.OR** from the ISPF Primary options panel, then select option **5** - SQL*Forms. Next, specify the ORACLE Subsystem ID of the ORACLE instance you want.

ORACLE SQL*MENU

SQL*Menu for MVS is an ORACLE application productivity tool that provides a single menu interface for running multiple data-processing tools.

SQL*Menu lets you create and execute menus from which you can carry out most of your data processing needs. A menu is a list of choices that correspond to tasks or commands. Selecting one of these choices calls an application, calls another menu, or issues commands. SQL*Menu works with all ORACLE software products, such as the ORACLE RDBMS, SQL*Forms, Oracle Reports, and SQL*Plus. Most products from other software vendors that work under MVS can also be invoked from SQL*Menu.

SQL*Menu Cataloged Procedures

Procedure	Description
GENMENU	Generate an executable menu file
DOCMENU	Document a SQL*Menu menu
RSCCNV	Create load module from an executable menu file

SQL*Menu Interactive Access

SQL*Menu may be built using the interactive menu design tool found on the ISPF ORACLE DEVELOPMENT TOOLS menu panel.

Select **=P.OR** then select option **4** – SQL*menu. Next, specify the ORACLE Subsystem ID of the ORACLE instance you want.

References

The following manuals may be of interest:

SQL*FORMS (Set of 5 books)
SQL*FORMS Designer Quick Reference
SQL*FORMS Operator Quick Reference
SQL*PLUS User Guide
SQL*PLUS User Guide & Reference
SQL*PLUS Quick Reference
RDBMS (Set of 4 books)
SQL QMX User Guide
SQL*NET 3270 User Guide
Pro*C User Guide
Pro*COBOL User Guide
Oracle Reports Reference Manual
Oracle Reports Building Reports
SQL*Forms Designer's Quick Reference
SQL*Forms Designer's Reference

SQL*Forms Designer's Tutorial
SQL*Forms Operator's Guide
SQL*Forms Operator's Quick Reference
SQL Language Reference
PL/SQL User's Guide and Reference
ORACLE for MVS User's guide
SQL*Menu User's Guide and Reference
ORACLE for MVS User's Guide
PL/SQL User's Guide and Reference

Generic Documentation (On-Line Documentation CD-ROM)

Concepts
Parallel Server Concepts and Administration
Error Messages
National Language Support Guide
Reference
Replication API Reference
SQL Reference
Supplied Packages Reference
SQL*Plus Quick Reference
SQL*Plus User's Guide and Reference
Intelligent Agent User's Guide
Administrator's Guide
Backup and Recovery Guide
Distributed Database Systems
Migration
Parallel Server Setup and Configuration Guide
Replication
Tuning
Utilities
Call Interface Programmer's Guide
Application Developer's Guide - Fundamentals
Application Developer's Guide - Advanced Queuing
Application Developer's Guide - Large Objects (LOBs)
Data Cartridge Developer's Guide
PL/SQL User's Guide and Reference
Pro*C/C++ Precompiler Programmer's Guide
Pro*COBOL Precompiler Programmer's Guide
Programmer's Guide to the Oracle Precompilers
SQL*Module for Ada Programmer's Guide

OS/390 specific documentation

Oracle for OS/390 Installation Guide
Oracle for OS/390 Message Guide
Oracle for OS/390 System Administration Guide
Oracle for OS/390 Tuning Guide
Oracle for OS/390 User’s Guide

USDA has the right to copy all ORACLE reference manuals as long as the ORACLE name remains on the documentation. All documentation is available on CD-ROM. The NITC has a limited subset of manuals that are available for agencies to produce their own copies. The agency ORACLE administrator should contact the Information Management and Customer Support Division at (816) 926-6681 to obtain a master set of documentation.

ORACLE url: http://www.oracle.com/

Chapter 2. DB2 Universal Database (UDB)

Introduction

IBM's DB2 is a Relational Database Management System (RDBMS) for the mainframe environment. In addition, it has expanded to include non-mainframe environments. DB2 can be installed on hardware ranging from a PDA to the largest mainframe. DB2 is built on open standards for ease of access and information sharing.

The following are some of the features of DB2 -

- Platform Independence
- Portability
- Scalability
- Availability
- Internet Architecture and Support
- ANSI standard SQL

Mainframe Access

DB2 can be accessed interactively by typing **=D.DB** at the ISPF Primary options panel. Interactive DB2 (DB2I) provides ISPF panel access to the following functions:

- SPUFI (SQL Processing Through File Input)
- Generation of Declare statements
- Compile and link edit of application programs
- Pre-compile of application programs
- Bind/Rebind/Free plans or packages
- Execute application programs
- Execute DB2 commands
- Execute DB2 utilities

Security

Authorization is required to access DB2 and must be requested in writing to NITC.

Security is maintained through the capabilities inherent in DB2 and security software installed on the DB2 platform.

Tools

NITC supports various IBM and third-party vendor software offerings that provide enhanced usage, productivity, monitoring, and maintainability for DB2. These tools are made available at the customer's request. Our customers currently use the following tools.

- DB2 Performance Monitor
- Query Management Facility (QMF)
- DB2 High Performance Unload
- BMC's Alter for DB2
- BMC's Activity Monitor for DB2
- CA's RC/Query
- CA's RC/Update
- CA's RC/Migrator

Cataloged Procedures

NITC provides procedures for compiling applications accessing DB2 and using a variety of programming languages. In addition, procedures are available for executing DB2 utilities:

Procedure	Description
DSNHASM	Preassemble, assemble and link edit an Assembler program
DSNHC	Precompile, compile and link edit a C program
DSNCOB2	Precompile, compile and link edit a Cobol II program
DSNHCPP	Precompile, compile and link edit a C++ program
DSNHCPP2	Precompile, compile, prelink and link edit a C++ program
DSNHFOR	Precompile, compile and link edit a Fortran program
DSNHICOB	Precompile, compile and link edit a MVS Cobol program
DSNHPLI	Precompile, compile and link edit a PLI program
DSNHSQL	Compile and link edit DB2 SQL Procedure program
DSNUPROC	Execute various DB2 Utilities

Batch SQL Execution

SQL can be executed in batch by using the TSO background program and either an application program or one of the sample programs DB2 provides. The following is an example of how to use the sample program DSNTEP2 (commonly referred to as batch SPUFI):

```
// SET SSID=DSN
//STEP01      EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB     DD   DSN=SYS30R.&SSID..DSNLOAD,DISP=SHR
//SYSOUT      DD   SYSOUT=*
//SYSTSPRT    DD   SYSOUT=*
//SYSPRINT    DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//SYSTSIN     DD   *
```

```
        DSN SYSTEM(DSN)
        RUN PROGRAM(DSNTEP2) -
              PLAN(DSNTEP2)
        END
//SYSIN   DD   *
        SELECT * FROM SYSIBM.SYSDATABASE;
```

Interfaces to DB2

NITC maintains other software products that interface with DB2:

- AS/DB2
- SAS/DB2
- FOCUS/DB2

More information on these interfaces may be found in the corresponding software chapter of this handbook.

Back Up and Recovery

NITC will maintain the archived logs from DB2 for 21 days. User agencies should be aware of their responsibility for maintaining image copies of their data. This should ensure that, along with the archived logs, all data is recovered. The greater the volatility of the data being maintained the greater the need for well planned back up and recovery procedures.

NITC can provide various levels of support for DB2 at the customer's request. These services include installation and administration of the DB2 subsystem/instance, database administration for application support, and application development. Contact NITC for more information.

References

DB2 and some third-party vendor manuals are available online and may be viewed in BookManager. For information on how to access and use BookManager, see the BookManager chapter of this handbook.

Additional reference materials may be obtained by accessing url: www.ibm.com and entering in the top right SEARCH box: **DB2**.

Chapter 3. FOCUS and Related Products

Introduction

FOCUS is a complete information control system with comprehensive features for entering, maintaining, retrieving, and analyzing data. FOCUS is designed for use by both users with no formal training in data processing and by data processing professionals who need tools for developing complete applications.

The non-procedural FOCUS language is designed to replace traditional programming languages in most application programming situations. Non-procedural languages free you from the constraints of specifying, in a predetermined way, how to process data. FOCUS takes you a level away from what the computer is doing from moment to moment, allowing you to concentrate on specifying what you wish to accomplish, such as print a report, update a file, create a graph, or build an entry screen.

Features

FOCUS provides the following features:

- The Report Writer, which works with existing FOCUS and non-FOCUS files.
- The Financial Modeling Language Facility for creating row-oriented financial reports.
- A full-screen text editor (TED) for creating and saving requests, Master Files, and other text files.
- The Dialogue Manager environment for building reusable FOCUS requests (FOCEXECs), including facilities for variable substitution, testing and branching, and reading from or writing to the terminal.
- The PC/Export Interface, used to extract specially formatted data (DIF, LOTUS, SYLK, WP) for use with other software products on Personal Computers.
- Facilities for designing menus and windows to select, enter, and display data.
- The data management facilities for loading and maintaining files.
- Facilities for designing full-screen data-entry forms, including two screen painters.
- An online, interactive full-screen database editor (FSCAN).

- FOCUS can access a wide range of file types and has interfaces to other types of databases. FOCUS also includes facilities for joining different types of files together that enable you to include data from several related databases in a report.

- WebFOCUS which provides web access to help you quickly build and deploy Web reporting and transactional systems over intranets, extranets, or the Internet.

Interactive FOCUS

FOCUS may be accessed by selecting option **=D.FO** from the ISPF Primary options menu or by typing **%FOCUS** from ISPF option 6 or the READY prompt.

FOCUS CLIST s

The following CLIST s may be used for online access to FOCUS and its interfaces:

FOCUS CLIST	Description
%FOCUS	Invoke the base FOCUS product
%FOCIDMS	Focus/IDMS Interface (10.2)
%FOCIDM12	Focus/IDMS Interface (12.0)
%FOCDB2	Focus/DB2 Interface
%FOCORACL	Focus/Oracle Interface

FOCUS Cataloged Procedures

The following cataloged procedures may be used for batch access to FOCUS and its interfaces:

Procedure	Description
KKFOCUS	Invoke the base FOCUS product
KKFOCIDT	Focus/IDMS Interface (10.2)
KKFOC12	Focus/IDMS Interface (12.0)
KKFOCDB2	Focus/DB2 Interface
KKFOCORA	Focus/Oracle Interface

KKFOCUS Procedure

```
//KKFOCUS  PROC  HILVL=
//FOCUS    EXEC PGM=FOCUS,REGION=4096K
//STEPLIB DD DSN=SYS90R.FOCUS.FOCLIB.LOAD,DISP=SHR
//USERLIB DD DSN=SYS90R.FOCUS.FUSELIB.LOAD,DISP=SHR
//ERRORS  DD DSN=SYS90R.FOCUS.ERRORS.DATA,DISP=SHR
//FOCSTACK DD UNIT=SYSDA,SPACE=(TRK,(20,20))
//FOCSORT  DD UNIT=SYSDA,SPACE=(TRK,(20,20))
//HOLDMAST DD UNIT=SYSDA,SPACE=(TRK,(10,10,10))
//HOLD     DD UNIT=SYSDA,SPACE=(TRK,(10,10))
//SAVE     DD UNIT=SYSDA,SPACE=(TRK,(20,20))
//FOCSML   DD UNIT=SYSDA,SPACE=(TRK,(20,20))
//SYSPRINT DD SYSOUT=*
//OFFLINE  DD SYSOUT=*
//MASTER DD DSN=&HILVL..MASTER.DATA,DISP=SHR
//       DD DSN=SYS90R.FOCUS.MASTER.DATA,DISP=SHR
//FOCEXEC DD DSN=&HILVL..FOCEXEC.DATA,DISP=SHR
//        DD DSN=SYS90R.FOCUS.FOCEXEC.DATA,DISP=SHR
//MODEL DD DSN=SYS90R.FOCUS.MODEL.DATA,DISP=SHR
```

HILVL - High-level qualifier of user data sets to be concatenated to FOCUS MASTER and FOCEXEC libraries.

Use JCL overrides to increase temporary space parameters, if necessary. Also, code additional file DD's to correspond to FOCUS execution requirements for the individual application.

You must include the FIN command at the end of SYSIN.

Sample JCL -

```
//job statement …
//STEP1 EXEC KKFOCUS,HILVL=yourhilvl
//MYDATA DD  DSN=my.file.focus,DISP=SHR
//SYSIN    DD  *
 TABLE FILE yourfile
 PRINT *
 END
 EX yourexec
 FIN
/*
```

WebFOCUS

WebFOCUS is a product driven by a high-powered reporting engine that allows you Web access to data on almost any platform in almost any file structure and dynamically generate it in a variety of formats, including HTML, XML, Excel, and Excel 2000.

If you wish to establish access to WebFOCUS, contact the NITC at 816-926-6681.

FOCUS Menu

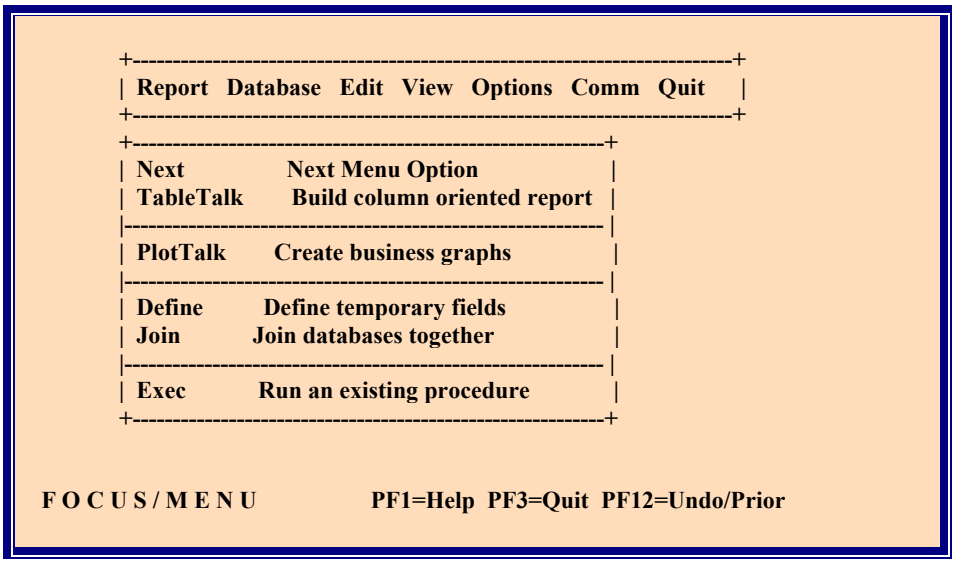
FOCUS Menu is a convenient way to access FOCUS facilities. Using FOCUS Menu you can access the Talk Technologies, use TED (the FOCUS editor), establish JOINS, do DEFINES, and access many other FOCUS features through a series of pop-up windows.

In order to use FOCUS Menu, data set SYS90R.FOCUS.FMU.DATA must be allocated to ddname FMU. Data set SYS90R.FOCUS.FOCEXEC.DATA must be allocated to ddname FOCEXEC.

Center FOCUS CLIST s have already been modified to concatenate the necessary data sets.
If you wish to modify your own agency CLIST s to include these file allocations, see data 1set KCUSER.FOCUS.CLIST S(FOCUS) for an example cist.

Usage

To execute FOCUS Menu, type **MENU** at the FOCUS prompt. The following FOCUS Menu panel will display:



The lower window contains Report Menu options. Selecting NEXT will move to the next Menu option.

To leave FOCUS Menu and exit to native FOCUS, press **PF3** and select the Command Level FOCUS option.

To leave FOCUS Menu and exit to TSO, press **PF3** and select the QUIT Focus option.

Be aware that when you type MENU to invoke FOCUS Menu, you are using a “LET” command. If you issue a TSO CLEAR ALL or establish your own MENU LET command, you will no longer be able to execute FOCUS Menu. To re-establish the MENU LET command, type the following at the FOCUS prompt:

LET MENU=EX FMMAIN

FOCUS Security

FOCUS security features offer security at every level, from the file itself down to specifying protection for specific values within fields. In addition, the FOCUS/CA-ACF2 Interface is an optional FOCUS facility that may be used to integrate the FOCUS security features with the CA-ACF2 file security system.

FOCUS/CA-ACF2 Interface

Although FOCUS, as it stands, respects all the security provisions of CA-ACF2, the addition of this interface will bring added benefits to FOCUS users without affecting the normal use of CA-ACF2 in any way.

The FOCUS ID set by the FOCUS/CA-ACF2 interface is not echoed to the terminal, and it is not shown in any batch job output listings. Many logon ids can be translated into the same FOCUS id. This makes it possible to create classes of FOCUS users, all of whom have different logon ids but the same level of access to FOCUS files.

FOCUS ids can be entered as overridable or non-overridable to restrict changes to passwords from within FOCUS.

Usage

The FOCUS/CA-ACF2 Interface reads the user ID used to gain access to the NITC computer and then checks the CA-ACF2 logon id record for an associated FOCUS id.

Once a user has been identified at logon time, he need not identify himself again to FOCUS.

To assign FOCUS id Agency security officers must request the FOCUS ID be added to each TSO/E LOGONID. Requests must be made in writing to the NITC security officer.

Simultaneous Usage (SU)

The SU feature of FOCUS allows many concurrent TSO/E and batch users to simultaneously update a FOCUS database. In "SU" mode, a batch-initiated job allocates one or more FOCUS databases to the "sync machine". This central job then controls all I/O, from both TSO/E ids and batch jobs.

Before attempting to implement an SU application, the Center suggests thorough consideration of the Simultaneous Usage Reference Manual (TSO Version).

Requirements

FOCUS requires that the following be in place in order to utilize the SU feature:

For each central database job, a communications data set must be established. FOCUS I/O will run in its own address space and this data set (FOCSU) provides the initial handshake between the source and the sync machines.

It also contains the common storage area (CSA) location where requests are queued, although subsequent transfer of commands and data takes place entirely in virtual storage. This data set must be allocated and cataloged on a permanently mounted volume.

The central database job must be able to write to this data set, and all source machines (TSO/E and batch) must have read permission. DCB attributes are now specified and space allocated should be minimal.

Example:

```
ATTRIB ADCB LRECL(16) RECFM(F B) BLKSIZE(16)
ALLOCATE DA('PREFIX.FOCSU.DATA') CATALOG SP(1) +
  TRACK USING (ADCB)
```

Prefix equates to an agency-established high-level qualifier and may be expanded to suggest the application for which SU is implemented.

JCL required for starting the sync machine:

```
//SYNCJOB EXEC PGM=HLISNK,REGION=1440K,PARM=' xxxx'
//STEPLIB  DD DSN=SYS90R.FOCUS.FOCLIB.LOAD,DISP=SHR
//FOCLIB   DD DSN=SYS90R.FOCUS.FOCLIB.LOAD,DISP=SHR
//FOCUSSU  DD DSN=SYS90R.FOCUS.FOCUSSU.FOCUS,DISP=SHR
//FOCSU    DD DSN=prefix.FOCSU.DATA,DISP=SHR
//MASTER   DD DSN=prefix.MASTER.DATA,DISP=SHR
//        DD DSN=SYS90R.FOCUS.MASTER.DATA,DISP=SHR
//FOCSORT  DD UNIT=SYSDA,SPACE=(TRK,(50,50))
//HLIPRINT DD SYSOUT=A,DCB=(RECFM=FB,LRECL=88,BLKSIZE=88)
                              OR
                    DCB=(RECFM=FB,LRECL=133,BLKSIZE=133)
//HLIERROR DD SYSOUT=A,DCB=(RECFM=FB,LRECL=88,BLKSIZE=88)
                              OR
                    DCB=(RECFM=FB,LRECL=133,BLKSIZE=133)
//DDNAME1  DD  DSN=DATA SETNAME1,DISP=SHR
//DDNAME2  DD  DSN=DATA SETNAME2,DISP=SHR
  .
  .
```

Prefix again refers to an agency high-level qualifier. The multiple DD statements refer to central FOCUS databases that will process under control of the sync machine.

The PARM field in the EXEC statement is optional and is used to activate the SU HLIPRINT facility. The syntax:

PARM=' ECHO' or **PARM=' STAT'**

ECHO requires an LRECL of 88 characters while STAT supplies 133 bytes of data. Information recorded pertains to user transactions and includes items such as user ID, filename in use, target segment for modify operations, type of activity and date/time stamps.

NOTE: DDNAME HLIPRINT may also be allocated to a sequential data set. However, it is not a transaction log and only provides a guide for restarting jobs that end prematurely. This ddname should not be a DD DUMMY statement.

This job should not time out and runs non-swappable in job class 'W'. Agencies are responsible for scheduling sync job execution.

In the individual FOCUS session, the communication data set (FOCSU) must be allocated, with a disposition of SHR, to a "syncid". (Please refer to SU Policy section of this chapter for syncid name requirements.) Then SU mode may be invoked by the following sequence:

```
USE FILENAME ON SYNCID
END
```

A user may access several databases controlled by different sync machines, as long as appropriate FOCSU files are allocated to syncids during the session. For example:

```
USE
ACCTREC ON SYNCA
ACCTPAY ON SYNCB
GENLEDG ON SYNITC
END
```

Following is the JCL required to stop the sync machine:

```
//SYNCDWN  EXEC  PGM=HLIKX
//STEPLIB   DD   DSN=SYS90R.FOCUS.FOCLIB.LOAD,DISP=SHR
//FOCSU     DD   DSN=prefix.FOCSU.DATA,DISP=SHR
```

The **?** **SU** FOCUS command will verify whether the session has successfully invoked SU mode. The most frequently encountered error conditions are:

(FOC542) Incorrect communication data set ddname in use or sync machine has not been started.

(FOC543) Sync machine does not have enough memory.

(FOC544) The sync has been shut down.

SU Policy

In order to establish a sync job application for an agency, the Center requires this information in writing (an entry in Info/Man will suffice). Please include:

1. A description of the application to be implemented.
2. A projected number of TSO/E users with access to the sync databases and maximum number of simultaneous users.
3. Indication if batch jobs will run against the sync machine.
4. An estimated average transaction rate.
5. Sync jobname and syncid.
6. Agency point of contact for questions concerning the sync job and/or application.

Starting and stopping the sync machine is the responsibility of the agency's designated point of contact. However, if an agency neglects to bring down the sync job cleanly prior to Center planned preventive maintenance, the Center reserves the right to cancel the sync job.

Do not assign a generic name to the syncid. In order to avoid duplication, please assign an agency abbreviation to the first three letters of the syncid (i.e. FSA, RD, OIG, etc.) and register the syncid with the Center's FOCUS point of contact prior to or at the time of the written request for SU.

FOCUS/ORACLE Interface

The FOCUS/ORACLE interface is operational and fully supports the Oracle relational data model. It provides the transaction processing and application development tools needed to update and maintain tables resident on the Oracle RDBMS.

The FOCUS/Oracle Interface is an additional layer of software responsible for translating FOCUS retrieval and update requests into an equivalent set of Oracle/SQL statements (the language supported by the Oracle RDBMS). Just as important, the Interface initiates and monitors communication between itself and the Oracle RDBMS, and provides descriptive error messages and recovery support when necessary.

The interface is comprised of two distinct components, "read" and "write". The read component translates FOCUS retrieval requests (e.g. TABLE and GRAPH) into Oracle/SQL statements that define the user's request to the Oracle RDBMS. The data returned by Oracle in response to these statements is then passed to the FOCUS report writer. The report writer is used for all FOCUS-readable files.

The "write" component acts in a similar fashion; it retrieves and maintains Oracle data through Oracle/SQL generated on behalf of standard FOCUS MODIFY requests submitted by the user.

Requirements

In order to access an Oracle table, you must first describe it in terms FOCUS can understand. The following elements must be in place before utilizing the interface:

1. A FOCUS master file descriptor, which tells FOCUS how to interpret the data stored in the individual fields (columns) of the Oracle table. It contains the field in the table and their data type and length. This member should be stored in the user's "MASTER" data set.

2. An access file descriptor, which provides FOCUS with information about the Oracle table itself. It includes the name of the Oracle table as it is known to the Oracle RDBMS, the table's primary key and, optionally, any column name that exceeds 12 characters. This member should be stored in the user's "FOCSQL" data set.

Both the Master and Access File Descriptions are created manually using your system editor or the FOCUS editor, TED. A subsequent release of the interface will allow the automatic creation of Master and Access File Descriptions for existing Oracle tables through the use of a utility program.

The master and access file descriptions must use the same name in order to link and define the Oracle table to FOCUS. Once you have a FOCUS master and access file description for the Oracle table, you can use the FOCUS facilities, such as Report Writer and Dialogue Manager, to handle your data. You do not need to know the SQL language.

If accessing Oracle databases interactively through FOCUS in a TSO/E session, the Oracle LOADLIB must be concatenated to ISPLLIB, which is the ISPF load library.

This can be accomplished in one of two ways:

1. The agencies signon CLIST can be modified (or established) to effect this concatenation. With this approach, the interface CLIST %FOCORACL must then be invoked from ISPF option 6.

2. The concatenation may be performed at the READY prompt. %FOCORACL must then be invoked from the READY using the following command:

   ISPSTART CMD(FOCORACL)

   The %FOCORACL CLIST allocates other system and user data sets; at run-time, the user must also include the Oracle subsystem by allocating the ddname ORA@ssn, where ssn is the subsystem name.

Usage

The Oracle Read/Write Interface operates in the MVS environment and runs interactively under the control of TSO or in batch mode. The Center CLIST , **%FOCORACL**, allocates the files required to invoke the interface.

The batch procedure **KKFOCORA** invokes execution of the Oracle interface to FOCUS in batch mode:

KKFOCORA Procedure

```
//KKFOCORA PROC HILVL=
//FOCUS    EXEC PGM=FOCUS,REGION=4096K
//STEPLIB DD DSN=SYS90R.FOCUS.FOCSQL.LOAD,DISP=SHR
//        DD DSN=SYS90R.FOCUS.FOCLIB.LOAD,DISP=SHR
//        DD DSN=SYS90R.FOCUS.FUSELIB.LOAD,DISP=SHR
//ERRORS  DD DSN=SYS90R.FOCUS.ERRORS.DATA,DISP=SHR
//FOCSTACK DD UNIT=SYSDA,SPACE=(CYL,(20,20))
//FOCSORT  DD UNIT=SYSDA,SPACE=(CYL,(30,30))
//HOLDMAST DD UNIT=SYSDA,SPACE=(TRK,(10,10,10))
//HOLD     DD UNIT=SYSDA,SPACE=(TRK,(20,20))
//SAVE     DD UNIT=SYSDA,SPACE=(TRK,(20,20))
//FOCSML   DD UNIT=SYSDA,SPACE=(TRK,(20,20))
//MODEL    DD DSN=SYS90R.FOCUS.MODEL.DATA,DISP=SHR
//SYSPRINT DD SYSOUT=*
//OFFLINE  DD SYSOUT=*
//MASTER DD DSN=&HILVL..MASTER.DATA,DISP=SHR
//       DD DSN=SYS90R.FOCUS.MASTER.DATA,DISP=SHR
//FOCEXEC DD DSN=&HILVL..FOCEXEC.DATA,DISP=SHR
//        DD DSN=SYS90R.FOCUS.FOCEXEC.DATA,DISP=SHR
```

Symbolic parameters:

HILVL - High-level qualifier of user data sets to be concatenated to the specified libraries.

FOCUS-DB2 Interface

The FOCUS-DB2 interface is operational and supports all FOCUS features for "READ-ONLY" access to data. Read access includes the FOCUS report writing features of TABLE, GRAPH, and ANALYSE; the Financial Reporting Language; and the JOIN and MATCH commands, which combine data from unrelated sources (DB2 tables and FOCUS databases) into one report.

Requirements

In order to access a DB2 table, you must first describe it to FOCUS in FOCUS terminology. The following elements must be in place before utilizing the interface:

1. A FOCUS master file descriptor, which identifies the DB2 table, the fields in the table and their data type and length. This member should be stored in the user's MASTER data set.

2. An access file definition, which serves as a link from FOCUS to DB2. It identifies the table creator, the tablename and dbspace name, and stores

information about the unique index and read access. This member should be stored in a data set allocated to FOCSQL.

The master and access file descriptions must use the same name in order to link and define the DB2 table to FOCUS. Once you have a FOCUS master and access file description for the DB2 table, you can use the FOCUS facilities, such as, report writer and Dialogue Manager, to handle your data. You do not need to know the SQL language.

Usage

The DB2 Read Interface operates in the MVS environment and runs interactively under the control of TSO/E or in batch mode. The Center CLIST , **%FOCDB2**, allocates the files required to invoke the interface.

The batch procedure **KKFOCDB2** invokes execution of the DB2 interface to FOCUS in batch mode:

KKFOCDB2 Procedure

```
//KKFOCDB2 PROC  HILVL=
//FOCDB2   EXEC PGM=FOCUS,REGION=4096K
//STEPLIB DD DSN=SYS90R.FOCUS.FOCLIB.LOAD,DISP=SHR
//       DD DSN=SYS90R.FOCUS.FOCSQL.LOAD,DISP=SHR
//       DD DSN=SYS90R.FOCUS.FUSELIB.LOAD,DISP=SHR
//ERRORS  DD DSN=SYS90R.FOCUS.ERRORS.DATA,DISP=SHR
//MASTER DD DSN=&HILVL..MASTER.DATA,DISP=SHR
//      DD DSN=SYS90R.FOCUS.MASTER.DATA,DISP=SHR
//FOCEXEC DD DSN=&HILVL..FOCEXEC.DATA,DISP=SHR
//       DD DSN=SYS90R.FOCUS.FOCEXEC.DATA,DISP=SHR
//FOCSQL  DD DSN=&HILVL..FOCSQL.ACCESS,DISP=SHR
//       DD DSN=SYS90R.FOCUS.FOCSQL.ACCESS,DISP=SHR
//FOCSTACK DD UNIT=SYSDA,SPACE=(CYL,(5,5))
//FOCSORT  DD UNIT=SYSDA,SPACE=(CYL,(5,5))
//HOLDMAST DD UNIT=SYSDA,SPACE=(TRK,(10,10,10))
//HOLD     DD UNIT=SYSDA,SPACE=(TRK,(20,20))
//SAVE     DD UNIT=SYSDA,SPACE=(TRK,(20,20))
//FOCSML   DD UNIT=SYSDA,SPACE=(TRK,(20,20))
//MODEL    DD DSN=SYS90R.FOCUS.MODEL.DATA,DISP=SHR
//OFFLINE  DD SYSOUT=*
```

Symbolic parameters:

HILVL - High-level qualifier of user data sets to be concatenated to the specified libraries.

AUTODB2

AUTODB2 is an interactive file generation facility designed to simplify the process of defining the DB2 tables to FOCUS. AUTODB2 produces both a master and access file description.

Usage

AUTODB2 is itself a FOCUS FOCEXEC. After entry into FOCUS, simply key **EX AUTODB2**. AUTODB2 will allocate temporary data sets and then present a screen requesting the following information:

1. The (same) name for the members to be placed on the master and access file partitioned data sets.

2. The DB2 table creator name (defaults to your TSO/E logonid.)

3. Name of the DB2 table that you are defining.

4. Level of functionality (select "R" for the read interface.)

5. Master file target is the name of the partitioned data set where the new master file description is to be placed. Default is yourhilvl.MASTER.DATA.

6. Access file target is the name of the partitioned data set where the new access file description is to be placed. Default is yourhilvl. FOCSQL.ACCESS.

During processing, if the data sets specified for the target files have not been created previously, AUTODB2 creates and catalogs them. If the data sets already exist, they will be allocated for use during the AUTODB2 procedure.

You can review the AUTODB2-generated file descriptions and edit them with your system editor or the FOCUS editor, TED. You can add optional attributes, such as, field titles, DEFINEd fields, print editing options, JOINs between tables (multi-table description), and FOCUS security.

FOCUS/CA-IDMS Interface

The FOCUS/CA-IDMS interface supports all FOCUS features for which "read-only" access to data is required, particularly printed report generation, statistical analysis, cataloged procedures, and use of the match command to combine data from several unrelated sources in one report.

Since FOCUS obtains data from CA-IDMS files through normal CA-IDMS read-only calls and requests only non-exclusive retrieval rights data integrity is not jeopardized.

The interface module enters and traverses the CA-IDMS data to retrieve selected records and present them, one at a time, to the FOCUS report writer. With proper allocations and CA-ACF2 security in place, queries using a CA-IDMS database can be as transparent as inquiries into a FOCUS database.

Additionally the interface supports discontiguous indexes, CALC fields, sorted set keys, and LRF cross-references.

Usage

FOCUS requires the following elements to be in place in order to utilize the IDMS interface:

A FOCUS master file descriptor (MFD), which contains the file name, segments, and field descriptions for the CA-IDMS database or for a particular hierarchical view of the database. (This member is stored in the user's "MASTER" data set.)

An access file definition, which indicates how each parent-child relation is implemented (as a set, via an index, or calc) and contains the CA-IDMS record area and set names needed to issue the appropriate calls. (This member should be stored by the user in a data set allocated to "FOCIDMS".)

Having the same name associates the master and access file descriptions. Both are coded in FOCUS syntax that, depending upon the size and complexity of the CA-IDMS database, can be a cumbersome task to define. For that reason, the NITC has installed "AUTOIDMS", an online FOCUS enhancement that will read CA-IDMS subschema information from the IDD and translate it into the FOCUS master and access file syntax required for the interface. (AUTOIDMS is discussed later in this chapter.)

Limitations

Some limitations exist for customers who wish to access CA-IDMS databases interactively through FOCUS. During the TSO/E session, the CA-IDMS LOADLIB must be concatenated to ISPLLIB, the ISPF load library. This can be accomplished in one of two ways:

1. The agency's sign on CLIST can be modified (or established) to effect this concatenation. With this approach, the interface CLIST FOCIDMS must be invoked from within ISPF, option 6.

2. The concatenation may be performed at the READY prompt; however, FOCIDMS must be invoked from the READY using the following command:

   ISPSTART CMD(FOCIDMS)

The batch procedure **KKFOCIDM** invokes execution of the IDMS interface to FOCUS in batch mode:

KKFOCIDM Procedure

```
//KKFOCIDM PROC  HILVL=
//FOCUS    EXEC PGM=FOCUS,REGION=4096K
//STEPLIB  DD DSN=SYS90R.FOCUS.FOCLIB.LOAD,DISP=SHR
//         DD DSN=SYS90R.FOCUS.IDMS.LOAD,DISP=SHR
//USERLIB  DD DSN=SYS90R.FOCUS.FUSELIB.LOAD,DISP=SHR
//ERRORS   DD DSN=SYS90R.FOCUS.ERRORS.DATA,DISP=SHR
//FOCSTACK DD UNIT=SYSDA,SPACE=(TRK,(20,20))
//FOCSORT  DD UNIT=SYSDA,SPACE=(TRK,(30,30))
//HOLDMAST DD UNIT=SYSDA,SPACE=(TRK,(10,10,10))
//HOLD     DD UNIT=SYSDA,SPACE=(TRK,(20,20))
//SAVE     DD UNIT=SYSDA,SPACE=(TRK,(20,20))
//FOCSML   DD UNIT=SYSDA,SPACE=(TRK,(20,20))
//MODEL    DD DSN=SYS90R.FOCUS.MODEL.DATA,DISP=SHR
//SYSPRINT DD SYSOUT=*
//OFFLINE  DD SYSOUT=*
//MASTER   DD DSN=&HILVL..MASTER.DATA,DISP=SHR
//         DD DSN=SYS90R.FOCUS.MASTER.DATA,DISP=SHR
//FOCEXEC  DD DSN=&HILVL..FOCEXEC.DATA,DISP=SHR
//         DD DSN=SYS90R.FOCUS.FOCEXEC.DATA,DISP=SHR
```

Symbolic parameters:

HILVL - High-level qualifier of user data sets to be concatenated to the specified libraries.

NOTE: You must supply JCL overrides to concatenate the CA-IDMS LOADLIB to STEPLIB. For CV mode, you must also include the file DD for the desired SYSCTL data set. To run local mode, you must supply the additional file DD's required to execute. It may be necessary to increase the temporary space parameters assigned in the procedure.

AUTOIDMS

AUTOIDMS is an interactive file generation facility that gathers subschema information from the IDD to produce FOCUS master file descriptions (MFDs) and their associated access file members. Once these files are present, queries may be built against CA-IDMS subschemas in native FOCUS or with TABLETALK, in batch or interactive mode.

Requirements

AUTOIDMS runs using the central version ONLY. The same online restrictions described in the CA-IDMS Limitations section are also applicable to AUTOIDMS, with an additional limitation.

When a CV contains multiple secondary IDDs, AUTOIDMS will only read/translate subschema information from the default IDD. (The default IDD is coded in the CA-IDMS schema.)

If translations from other than the default IDD are needed, an additional step is required. Member IDMSIDD is a "generic" FOCUS access file descriptor (AFD) for the IDMS network subschema.

This member must be copied from SYS90R.FOCUS.FOCIDMS.DATA into the user's AFD. Then, in order to point AUTOIDMS to the proper dictionary, the first line of the member must be modified from:

SSCHEMA=IDMSNWKA,RELEASE=12,$

to:

SSCHEMA=IDMSNWKA,RELEASE=12,DBNAME=xxxxx,$
(xxxxx is the desired dictionary name)

Again, the CA-IDMS LOADLIB must be concatenated to ISPLLIB, and agencies may wish to screen usage of this feature to specific or limited number of logonids. Several other required allocations are accomplished in the CLIST **%FOCIDMS**, including the user's access file (DDNAME: FOCIDMS).

(This data set is concatenated to the Center's FOCIDMS data set, where the IDMSIDD member provides only for pointing to the default IDD for the schema.) Finally the user must allocate the SYSCTL data set, either via TSO/E commands, in the FOCUS profile, or in a FOCEXEC.

Usage

AUTOIDMS is a FOCUS FOCEXEC. After entry into FOCUS, simply key **EX AUTOIDMS**.

AUTOIDMS will allocate some temporary data sets and then present a screen requesting the following information:

1. The CA-IDMS subschema name for which the master and access file descriptions will be produced.

2. The (same) name for the members that will be generated and placed on the master and access file partitioned data sets.

3. Name of the target master file PDS (will default to the user's hilvl..MASTER.DATA).

4. Name of the target access file PDS (will default to the user's hilvl.FOCIDMS.DATA).

   NOTE: Items 3 and 4 may be overwritten, but the user must have CA-ACF2 write authority for the entries and they must be allocated to DDNAMES "MASTER" and "FOCIDMS".

5. Indication if the descriptions are new or should overlay current members.

At this point, the IDD is checked and a screen of all the records defined to the subschema is presented. The user must select one (and only one) "ROOT" record for the FOCUS master and it is generally a record with a calc and/or index field.

The user selects the record by positioning the cursor under the '*' on the line for the applicable record and pressing PF6. Then pressing PF3 will "kill" the report and AUTOIDMS begins processing.

The user then selects the record/set combinations that are coded as FOCUS descendents.

Limitations

The following limitations exist in AUTOIDMS:

1. AUTOIDMS chooses the first subschema with the requested name from the IDD, so different versions with the same name cannot be differentiated.

2. AUTOIDMS does not prompt the user to enter the desired IDD name, nor does it automatically code the DICTNAME into the access file description (AFD). To allow online inquiries that utilize IDDs besides the default IDD, the AFD must be revised to reflect the following:

   SSCHEMA=subschema name,RELEASE=12,DICTNAME=name of IDD,$

You may receive the error "REQUESTED MODULE IDMS NOT FOUND" for several reasons:

1. The CA-IDMS load library has not been properly concatenated to ISPLLIB.

2. Invocation of FOCIDMS is inconsistent with the manner in which the CA-IDMS loadlib has been concatenated to ISPLLIB.

3. The CV is not active.

4. The error "REQUESTED MODULE IDMSR NOT FOUND" indicates an error has occurred during the allocation of the access (FOCIDMS) files.

FOCUS COBOL FD Translator

The FOCUS COBOL FD Translator is a productivity tool that builds a FOCUS Master File Descriptor (MFD) from a COBOL File Descriptor (FD). The Translator may be invoked in batch or interactive mode.

Interactive Mode (TSO/E):

1. In the ISPF Editor, extract the File Descriptor from the COBOL program. Include only the elementary and group items. Omit the FD and associated statements.

2. From TSO option 6 or the READY, type **%FOCCTF**. At the FOCUS prompt (>>), type **ex COBFD**.

3. The Translator prompts for the dataset/member name information, MASTER target data set, and processing options.

4. The Translator generates and stores the FOCUS Master File Descriptor and produces a hardcopy report in process.

Batch Mode:

Example JCL:

```
//jobname…..
//CTFJOB  EXEC PROC=KKFOCCTF, FUNC=REPL,
//    MASTER='hilvl.MASTER.DATA',
//    SKPDASH=1
//SYSIN   DD  DSN=source.library(coboldescriptor),  DISP=SHR
/*
```

A COBOL FD Translator Users Manual is available from IBI, Inc. See the References section of this chapter for ordering information.

FOCUS Training

There are many online training courses available for FOCUS. See Teleview NEWS, Option 2 – Computer Based Training, for more information regarding online training or contact the NITC Customer Support Center at 816-926-6681 for assistance.

Information Builders, Inc. also provides a complete spectrum of FOCUS training for both data processing professionals and end-users. To supplement the online CBT courses, IBI offers classes at its branch offices and will arrange on site training, if desired. Course curriculum and descriptions may be obtained from IBI or by contacting the NITC Helpdesk at (816) 926-6681.

References

The following FOCUS reference manuals are suggested for use with the products installed at NITC:

FOCUS for S/390 Manual Set
Simultaneous Usage Reference Manual for TSO
A Guide to FOCUS Database Design
IDMS/R Interface Users Manual
Oracle Interface Users Manual
DB2 and SQL/DS Read/Write Interface Users Manual
ACF2 Interface Users Manual
FOCUS for IBM Mainframe Talk Technology User Manual
COBOL FD Translator Users Manual

To order manuals, please send purchase orders to:

Information Builders, Inc.
2 Penn Plaza, 24th Floor
New York, NY 10121-2898

(212) 736-4433 ext. 3396

A variety of technical information is also available at url: http://www.ibi.com/

Chapter 4. CA-IDMS and Related Products

Introduction

CA-IDMS/DB (Integrated Database Management System) software from Computer Associates International is a relational and network DBMS that allows users to model their application requirements with a minimum of technical complexity.

The system offers a variety of options, among them a powerful central version, a data communications package and a personal computer version. CA-IDMS/DB combines both network and relational DBMS architectures in one system. The following is a brief description of the additional program productivity tools and options for CA-IDMS/DB software.

CA-IDMS/DB: CA-IDMS/DB is a high performance network and relational database management system that centralizes and controls the data resources. It supports applications for diverse data processing environments and provides easy-to-use tools to make data accessible.

CA-IDMS/CV: CA-IDMS/CV (Central Version) allows multiple batch and/or online applications to access and update a CA-IDMS/DB database. All applications using central version services share the same copy of the CA-IDMS/DB database management system and benefit from CA-IDMS/DB security and procedures that control against concurrent update.

CA-IDD: CA-IDD (Integrated Data Dictionary) is used to control and report on information stored centrally in the data dictionary.

The major components are described below:

1. DDDL (Data Dictionary Definition Language) compiler accepts DDDL source statements, generates diagnostic error messages, and updates the dictionary/directory.

2. DDR (Dictionary/Directory Reporter) generates reports from information stored in the dictionary/directory. DDR produces both standard and special purpose reports.

CA-OLM: CA-OLM (Online Mapping) is the online facility used to define and generate maps. A map is a formatted terminal screen image used to communicate between an application and a terminal operator.

CA-ADS: CA-ADS (Application Development System) is a programming productivity tool (including a fourth-generation language) that allows customers to develop and execute online applications. It enables applications developers to query and update a CA-IDMS/DB database.

CA-ADS consists of three major components, as follows:

1. ADSA (ADS/Online Application Generator) is an application design and prototyping tool. The application generator allows the application developer to define the components and structure of an application and to specify the run-time flow of control.

2. ADSG (ADS/Online Dialog Generator) creates dialog load modules from definitions stored in the data dictionary. The dialog generator allows the application developer to define the structure of a dialog and add components to the dialog.

3. CA-ADS (ADS/Online Application) is a named set of related functions or dialogs used to perform a specific business task, such as inventory or payroll. An application provides a structure for the execution of the task as well as a means of navigation through the structure.

CA-OLQ: CA-OLQ (On-Line Query) is used by application developers and customers to view and report on contents of a CA-IDMS/DB database. CA-OLQ can also be used to create, update, delete tables.

CA-CULPRIT: CA-CULPRIT is the informational retrieval tool that generates customer-designed reports from CA-IDMS/DB databases and other, external files.

EDP AUDITOR: EDP AUDITOR is a comprehensive library of CA-CULPRIT routines that perform auditing tasks, such as confirmations, file footing, exception and summary analysis, and sampling.

Policy

The NITC subscribes to Computer Associates' service TOTAL CLIENT CARE (TCC) that provides the Center with online PTF's, documentation update, and technical notes as soon as they are reviewed and approved (before publication). Any emergency PTF can be downloaded to a PC and then uploaded to the mainframe for quick problem resolution.

General-purpose test and production central version (CV) systems are available to customers with limited database processing needs. NITC will provide technical Data Communication Administration (DCA) services. However, agencies are expected to provide their own Database Administrators (DBA).

The Center will assist customers with design reviews of their database management systems and is willing to assist customers with fine tuning their systems. Reviews may be arranged by phone at (816) 926-6681 or in writing to the Information Management and Customer Support Division.

Requests for a central version operation may be made in INFO/MAN (See INFO/MAN chapter of this Handbook) or by memo addressed to the Chief of the Information Management and Customer Support Division at the NITC.

All NITC CA-IDMS/DC systems run non-swappable. System security requires that NITC own all APF libraries. Agency Security Officers must contact the NITC Security Staff to request access to the authorized libraries.

No customer application modules are allowed in the authorized libraries. Unique CV and SVC numbers will be assigned by the Center.

VTAM application id's (APPLID) associated with the CA-IDMS/DC system must contain a unique agency identifier in the first 3 positions.

CA-IDMS/DC startup module names must begin with "IDMS" followed by a 2-digit number assigned by the Center. Conforming to these naming conventions and standards ensures that no CA-IDMS/DC is in conflict with one from another agency.

It is the Center's policy to support the current maintenance tape with the latest PTFs applied and the prior final release maintenance tape.

This support policy is consistent with Computer Associates release support. Only required PTFs that pertain to the Center's environment will be applied. Optional PTFs will be reviewed for minimal impact on our customers before being applied.

The Center has developed a procedure to notify all customers when PTFs are applied to any CA-IDMS releases that are supported. When a PTF is downloaded from TCCS, the description is stored in a data set that the customer can access KCUSER.XXXXXX.PTFS) where XXXXXX is the CA-IDMS release). Members of the dataset will be named as follows:

xyaabbbb where **x** = P (PTF)
O (Optional PTF)
T (Technical Note)

y = second digit of the year
aa = month of PTF
bbbb = PTF number

Examples:

P9012345 would be PTF 89-01-2345

O9123456 would be Optional PTF 89-12-3456

All CA-IDMS customers are notified by email that a PTF has been applied. The customer should copy the entire load library to assure that there are no partial or incomplete PTFs.

Security

Each agency's Security Officer must request DBA authority for specific logon ids in writing to the Center's Security Officer. DBA authority should be granted only to personnel responsible for database administrative functions.

CA-ICMS

Information Center Management Systems (CA-ICMS) is the link between the PC and the mainframe that provides you with direct access to live production data.

CA-ICMS consist of the following components:

IDB MANAGER: The tool that describes, organizes, and controls all customers and information in the ICMS environment. IDB Manager consists of a series of menu-driven screens that are used to create and maintain corporate and private catalogs.

IDB COMMUNICATIONS: The communications component of CA-ICMS that links PCs and Departmental Computers to the IBM mainframe.

IDB MAIL FACILITY: The facility that controls the exchange of electronic mail among PC customers.

CA-ICMS Optional Tools

The following optional tools are available from Computer Associates (CA) for use with CA-ICMS:

1. CA-Goldengate -- A set of integrated software tools for the PC, and the Goldengate information link, the software that creates a direct connection between Goldengate and ICMS.

   When the CA-Goldengate information link is installed, CA-Goldengate customers can:

   - Access mainframe information as if it were stored on the PC.
   - Generate information at the PC and upload this information to the mainframe.
   - Perform relational processing on CA-ICMS data tables.
   - Send information to other CA-Goldengate customers through the IDB mail facility.

2. CA-Infogate -- A subset of CA-Goldengate designed to work with CA-ICMS to provide customers with access to the Corporate information required for PCs.

CA-Infogate performs three primary functions for customers:

3. Provides customers of IBM-PC compatible software packages with access to mainframe data and information.

Allows a customer to exchange information easily, regardless of their physical location or the format of the data.

Acts as a system administrator, or shell, that integrates different PC programs while managing both micro and mainframe information.

CA-Culprit

CA-CULPRIT is a parameter driven report generator for use in the IBM and equivalent computer environments. It allows data retrieval in a format that meets specific requirements.

Cataloged Procedures

Procedure	Description
KKCULPRT	CA-Culprit

Example Execution of CA-Culprit -

```
//STEP01   EXEC KKCULPRT,LIB='COPIED SOURCE LIBRARY'
//SYS010   DD  DSN=INPUT FILE
//SYS020   DD  DSN=OUTPUT FILE


//DICTDB   DD  DSN=DICTIONARY      *** LOCAL MODE
//SYSCTL   DD  DSN=SYSCTL FILE     *** CV MODE


//SYSIN    DD  *
INPUT CARDS
//*
```

NOTE: Other SYSnnn DD statements may be added in the appropriate sequence of the execute JCL. See a CULPRIT Operations Manuals for details. Information on obtaining reference manuals is in the Reference section of this chapter.

If any of the work files are not large enough for your application please use JCL overrides to increase space parameters.

CA-UNIPACK/DBA

The CA-UNIPACK/DBA products in this unit provide software tools to manage, control, tune, and monitor the CA-IDMS/DB environment. The following software is included in this package:

CA-IDMS/ADS ALIVE: CA-IDMS/ADS Alive is a source-level testing and debugging tool that provides the CA-ADS developer with complete control over the execution of the CA-ADS environment. CA-ADS/Alive is used to assist the developer in correcting coding and design errors, as well as, the database analyst in monitoring database errors.

CA-IDMS/DB ANALYZER: DB/Analyzer is a parameter-driven utility that analyzes the physical organization of a CA-IDMS database. The utility produces a series of analytical and statistical reports necessary for both planning and assessing database reorganization. With DB/Analyzer the customer can analyze an entire database or portions of a database. DB/Analyzer also allows the customer to compare current physical organization conditions with past ones, or two sets of past conditions with each other. The utility provides a method to forecast and monitor database growth.

CA-IDMS/DB AUDIT: DB/Audit is a utility that allows customers to audit the physical integrity of all or part of a CA-IDMS database. In addition, the customer can direct DB/Audit to fix most of the errors found and to report the errors and corrections. A report of simulated corrections can be examined before actually updating the database. Elimination of physical inconsistencies in the database allows smooth operations for the growth of applications development. DB/Audit also physically deletes LDELs (Logical Deleted occurrences). It is the only means to determine orphan records in a database without the cost of running the 'walk-next' option.

CA-IDMS/DB REORG: DB/Reorg is a utility used to reorganize a database without unloading and reloading the entire database. DB/Reorg enables the customer to reorganize one or more areas at a time. DB/Reorg also provides the capability to control the placement of records to insure that the most frequently accessed records are placed on their target page and the less accessed records overflow if necessary. It is possible to focus on specific database organization problems and reorganize only those areas that are causing performance problems. DB/Reorg resolves all set pointers affected by changing the placement of records, particularly those in areas not being reorganized.

CA-IDMS/DML ONLINE: DML/Online is a CA-IDMS programmer productivity tool designed to allow immediate online execution of all data manipulation language (DML) commands. DML/O accelerates new system development, supports existing applications and helps train programmers who are inexperienced in CA-IDMS DML/O syntax. Both new and experienced programmers can learn to navigate quickly through a database with DML/O.

The software supports all CA-IDMS retrieval, modification, save and control commands. As a training aid, DML/O is an easy resource to teach essential IDMS concepts such as currency to new programmers. With DML/O, programmers can build and test database data without writing additional load programs. Programmers do not need to learn a new DML syntax in order to execute DML/O, since the same DML syntax used to write CA-IDMS programs is used in DML/O.

CA-IDMS/SCHEMA MAPPER: Schema mapper is a parameter driven utility which automatically creates Bachman diagrams from CA-IDMS schemas and subschemas stored in the data dictionary. Schema mapper also produces a transfer file which retains the expanded parameter input statements for future processing, a cross-reference report which includes descriptions of sets and indices from the schema mapper execution, and an audit report which contains a summary of all processing and various messages and warnings generated during program execution.

The following members in 'KCDBMS.S14100.INSTLLIB' contain description, CA-IDMS/DC installation, execution, and model JCL where applicable:

Member Name	Product
ADSALIVE	ADS/Alive
DBANLYZ	DB/Analyzer
DBAUDIT	DB/Audit
DBREORG	DB/Reorg
DMLO	DML/O
SCHMPER	Schema Mapper

NOTE: Please note that only DBA types with CA-ACF2 read permissions to KCDBMS high-level qualifier data sets can install the CA-UNIPACK/DBA products.

The above members are set up with print control statements to be printed by ICEGENER. Code the SYSUT1 and SYSUT2 DD statements as follows:

```
//SYSUT1   DD DSN=SYS90R.KCDBMS.INSTLLIB(ADSALIVE),DISP=SHR
//         DD DSN=SYS90R.KCDBMS.INSTLLIB(DBANLYZ),DISP=SHR
//         DD DSN=SYS90R.KCDBMS.INSTLLIB(DBAUDIT),DISP=SHR
//         DD DSN=SYS90R.KCDBMS.INSTLLIB(DBREORG),DISP=SHR
//         DD DSN=SYS90R.KCDBMS.INSTLLIB(DMLO),DISP=SHR
//         DD DSN=SYS90R.KCDBMS.INSTLLIB(SCHMPER),DISP=SHR
//SYSUT2 DD SYSOUT=A,DCB=(LRECL=80,BLKSIZE=3120,RECFM=FBA)
```

CA-UNIPACK/DBA

The CA-UNIPACK/DBA products in this unit provide software tools to manage, control, tune, and monitor the CA-IDMS/DB environment. The following software is included in this package:

CA-IDMS/ADS ALIVE: CA-IDMS/ADS Alive is a source-level testing and debugging tool that provides the CA-ADS developer with complete control over the execution of the CA-ADS environment. CA-ADS/Alive is used to assist the developer in correcting coding and design errors as well as the database analyst in monitoring database errors.

CA-IDMS/DB ANALYZER: DB/Analyzer is a parameter-driven utility that analyzes the physical organization of a CA-IDMS database. The utility produces a series of

analytical and statistical reports necessary for both planning and assessing database reorganization. With DB/Analyzer the customer can analyze an entire database or portions of a database. DB/Analyzer also allows the customer to compare current physical organization conditions with past ones, or two sets of past conditions with each other. The utility provides a method to forecast and monitor database growth.

CA-IDMS/DB AUDIT: DB/Audit is a utility that allows customers to audit the physical integrity of all or part of a CA-IDMS database. In addition, the customer can direct DB/Audit to fix most of the errors found and to report the errors and corrections. A report of simulated corrections can be examined before actually updating the database. Elimination of physical inconsistencies in the database allows smooth operations for the growth of applications development. DB/Audit also physically deletes LDELs. (Logical Deleted occurrences). It is the only means to determine orphan records in a database without the cost of running the 'walk-next' option.

CA-IDMS/DB REORG: DB/Reorg is a utility used to reorganize a database without unloading and reloading the entire database. DB/Reorg enables the customer to reorganize one or more areas at a time. DB/Reorg also provides the capability to control the placement of records to insure that the most frequently accessed records are placed on their target page and the less accessed records overflow if necessary. It is possible to focus on specific database organization problems and reorganize only those areas that are causing performance problems. DB/Reorg resolves all set pointers affected by changing the placement of records, particularly those in areas not being reorganized.

CA-IDMS/DML ONLINE: DML/Online is a CA-IDMS programmer productivity tool designed to allow immediate online execution of all data manipulation language (DML) commands. DML/O accelerates new system development, supports existing applications and helps train programmers who are inexperienced in CA-IDMS DML/O syntax. Both new and experienced programmers can learn to navigate quickly through a database with DML/O.

The software supports all CA-IDMS retrieval, modification, save and control commands. As a training aid, DML/O is an easy resource to teach essential IDMS concepts such as currency to new programmers. With DML/O, programmers can build and test database data without writing additional load programs. Programmers do not need to learn a new DML syntax in order to execute DML/O, since the same DML syntax used to write CA-IDMS programs is used in DML/O.

CA-IDMS/SCHEMA MAPPER: Schema mapper s a parameter driven utility that automatically creates Bachman diagrams from CA-IDMS schemas and subschemas stored in the data dictionary.

Schema mapper also produces a transfer file which retains the expanded parameter input statements for future processing, a cross-reference report which includes descriptions of sets and indices from the schema mapper execution, and an audit report which contains a summary of all processing and various messages and warnings generated during program execution.

The following members in 'KCDBMS.S14100.INSTLLIB' contain description, CA-IDMS/DC installation, execution, and model JCL where applicable:

Member Name	Product
ADSALIVE	ADS/Alive
DBANLYZ	DB/Analyzer
DBAUDIT	DB/Audit
DBREORG	DB/Reorg
DMLO	DML/O
SCHMPER	Schema Mapper

NOTE: Please note that only DBA types with CA-ACF2 read permissions to KCDBMS high-level qualifier data sets can install the CA-UNIPACK/DBA products.

The above members are set up with print control statements to be printed by ICEGENER. Code the SYSUT1 and SYSUT2 DD statements as follows:

```
//SYSUT1 DD DSN=SYS90R.KCDBMS.INSTLLIB(ADSALIVE),DISP=SHR
//         DD DSN=SYS90R.KCDBMS.INSTLLIB(DBANLYZ),DISP=SHR
//         DD DSN=SYS90R.KCDBMS.INSTLLIB(DBAUDIT),DISP=SHR
//         DD DSN=SYS90R.KCDBMS.INSTLLIB(DBREORG),DISP=SHR
//         DD DSN=SYS90R.KCDBMS.INSTLLIB(DMLO),DISP=SHR
//         DD DSN=SYS90R.KCDBMS.INSTLLIB(SCHMPER),DISP=SHR
//SYSUT2 DD SYSOUT=A,DCB=(LRECL=80,BLKSIZE=3120,RECFM=FBA)
```

CA-Unipack/Developer

The CA-UNIPACK/DEVELOPER is a collection of software tools designed to maximize a developer's efficiency. The following products make up the CA-UNIPACK/DEVELOPER package:

CA-IDMS/ADS TRACE: ADS/Trace is an interactive debugging interface for ADS/O dialogs. ADS/Trace automatically enters trace code in the ADS/O dialog by specifying trace options from a special screen. Likewise, it also deletes the code after the debugging session is complete. This is accomplished by turning off the trace options from the same screen mentioned above.

After specifying the trace options, it is necessary to re generate the dialog with the addition of the ADS/Trace work record. Again, it is necessary to return to ADSG to remove the ADS/Trace work record after the debugging session is complete. ADS/Trace displays the value of any dialog work record, database field, or map variable by manually specifying the exhibit verb in the source code.

ADS/Trace also displays literals by manually specifying 'LITERAL' and the actual literal in the source code. Consult the manual for details. During the execution of ADS/Trace,

command processing and DML error status are captured and highlighted on the trace display.

The trace can then be viewed, printed or saved in a queue for a replay at a later time. ADS/Trace allows for up to 100 replays for the same dialog and is designated by a unique replay number so that each execution of logic can be reviewed as needed. Each time the dialog is executed with the trace options on, a new number is assigned to the replay of each process until the 100 replays per dialog is reached. ADS/Trace preserves the logical sequence in the replay without any rollback.

CA-IDMS/DICTIONARY MODULE EDITOR (DME): DME allows customers to modify existing modules in the primary and/or secondary dictionaries. It is necessary for the DBA or Dictionary Administrator to initially add a dummy module for each process or module the customer will access.

Customers can only modify; they cannot delete or add modules. Selection is based on class and attribute structure or module name. The class and attribute selection screen provides a list from which to choose.

It is ideal for ADS/O programmers who have been denied access to IDD, only because they would have access to other entities. DME uses ISPF functions (full screen edit functions but without home key, split screen and line indention capabilities).

CA-IDMS/DC SORT: Formerly called TP/Sort, DC/Sort is an online sort utility for CICS CA-IDMS/DC environments. It provides a means to sort data online, regardless of file structure and original sequence. DC/Sort eliminates the need to design and maintain specific sort sequences in your files or on your database.

The sort criteria can be elected at run-time or at compile time by coding the appropriate DC/Sort syntax in the ADS/O dialog or CA-IDMS/DC COBOL program. One terminal can support up to 20 concurrent sort sessions with each terminal using up to 16 sort keys.

CA-IDMS/ENFORCER: The Enforcer provides automatic verification and enforcement of naming standards, entered into the Integrated Data Dictionary (IDD). You can maximize the benefit of establishing these standards through the use of the Enforcer.

CA-IDMS/DICTIONARY MIGRATOR: Dictionary Migrator is a system migration tool for transferring complete systems or portions of a system from one dictionary to another. It is especially helpful in an ADS/O environment. Dictionary Migrator extracts from the dictionary the source (syntax) that will add all ADS/O entities, elements, records, dialogs, messages, tables, schemas, subschemas, maps, panels, and load modules to an object dictionary.

Migrator also provides comprehensive reports before and after the migration. It is a parameter-driven utility that uses work syntax files created by dictionary migrator to populate an object dictionary.

DICTIONARY MIGRATOR ASSISTANT (DMA): DMA is an interactive tool used to facilitate coding input parameters and JCL to submit Dictionary Migrator for execution. DMA saves multiple versions of parameters and JCL in parm files associated with the customer's ID. Customers can submit Dictionary Migrator jobs to OS/MVS via CA-IDMS/DC for execution in either local mode or under CV depending on the parameters coded.

DMA is considered a part of the Dictionary Migrator described above, and is not a separate function of the CA-UNIPACK/DEVELOPER.

CA-IDMS/DATABASE EXTRACTOR (DBX): DBX (formerly Test Database Builder (TDB)) is an applications development tool which significantly speeds up the testing and maintenance phases of applications development by reducing or eliminating the need to develop special test database load programs.

MASTERKEY: Masterkey is a task initiation control facility for IDMS-DC. By controlling the invocation of tasks, the developer or database administrator is able to save keystrokes, streamline workflow, and provide a certain amount of security in the IDMS environment.

NOTE: Only DBA types with CA-ACF2 read permissions to KCDBMS high-level qualifier data sets can install the developer toolkit.

The following members in KCDBMS.S14100.INSTLLIB contain description, CA-IDMS/DC installation, execution, and model JCL where applicable:

Member Name	Product
ADSTRACE	ADS/Trace
DBX	DME
DICMIG	Dictionary Migrator
DCSORT	DC/Sort
ENFORCER	Enforcer
MASTRKEY	Masterkey

The members are set up with print control statements to be printed by ICEGENER. Code the SYSUT1 and SYSUT2 DD statements as follows:

```
//SYSUT1 DD DSN=SYS90R.KCDBMS.INSTLLIB(ADSTRACE),DISP=SHR
//          DD DSN=SYS90R.KCDBMS.INSTLLIB(DBX),DISP=SHR
//          DD DSN=SYS90R.KCDBMS.INSTLLIB(DICMIG),DISP=SHR
//          DD DSN=SYS90R.KCDBMS.INSTLLIB(DME),DISP=SHR
//          DD DSN=SYS90R.KCDBMS.INSTLLIB(DQF),DISP=SHR
//          DD DSN=SYS90R.KCDBMS.INSTLLIB(DCSORT),DISP=SHR
//          DD DSN=SYS90R.KCDBMS.INSTLLIB(ENFORCER),DISP=SHR
//          DD DSN=SYS90R.KCDBMS.INSTLLIB(MASTRKEY),DISP=SHR
//SYSUT2 DD SYSOUT=A,DCB=(LRECL=80,BLKSIZE=3120,RECFM=FBA)
```

CA-UNIPACK/ANALYZER

The CA-UNIPACK/ANALYZER is a collection of software tools from Computer Associates. The following products make up the CA-UNIPACK/ANALYZER package:

CA-IDMS/PERFORMANCE MONITOR: The CA-IDMS/Performance Monitor is a three-part performance and tuning tool that allows the database administrator the ability to monitor hardware and software resource utilization in a CA-IDMS- DC/UCF system.

The Realtime Monitor captures and displays information describing the use of system resources. The Realtime Monitor is either a conversational or pseudo-conversational task. In either case, it automatically refreshes the screen with current statistics. This information is drawn directly from runtime control blocks maintained by the CA-IDMS-DC/UCF system at the time of the request.

The Interval Monitor captures system-wide wait-time statistics and information related to wait-time statistics for each interval. An interval is a unit of time (30 minutes, 60 minutes, etc.). The time spanned by each interval is established by the system administrator, as is the total number of intervals to be maintained.

The Application Monitor continuously captures and records task information, and reports that information either on-line or through batch reports.

The Performance Monitor comes equipped with a number of canned culprit report modules (PREPORTS). The reports can be generated from the CA-IDMS log dumps or from SMF data. If the SMF option is chosen, special arrangements must be made with the Information Management and Customer Support Division to get the data extracted from SMF.

The Interval and Application Monitors require an assembly of the #PMOPT module.

CA-IDMS/JOURNAL ANALYZER: Journal Analyzer provides the customer with a method to extract data from the CA-IDMS archive journals. Journal Analyzer is flexible enough to tailor the extracted information to the customer's specific needs. Summary reports are prepared in a variety of formats. The reports enable the customer to monitor database performance and identify problem areas. Also Journal Analyzer displays before and after images of updated records (data and prefix) on the archive journals.

CA-IDMS/LOG ANALYZER: Log Analyzer is a utility that allows the customer to run performance and statistical oriented reports, similar to those of the Journal Analyzer reports. Log Analyzer produces billing-related reports that can be used as a basis for charge back of CA-IDMS resources.

In addition, Log Analyzer provides a facility to produce a billing file for subsequent input into billing or accounting systems.

CA-IDMS/TASK ANALYZER: Task Analyzer is a CA-IDMS/DC task reporting utility. Using multiple CA-IDMS exits, Task Analyzer gathers statistics on programs,

ADS/O dialogs, tasks, integrated indexes and writes these statistics to the CA-IDMS log (or optionally, the SMF file).

The customer can generate a number of reports with a variety of selection criteria.

The following members in KCDBMS.S14100.INFO contain description, CA-IDMS/DC installation, execution, and model JCL where applicable:

Member Name	Product
JRNANLYZ	Journal Analyzer
LOGANLYZ	Log Analyzer
TASKNLYZ	Task Analyzer

NOTE: Only DBA types with CA-ACF2 read permissions to KCDBMS high-level qualifier data sets can install the CA-UNIPACK/ANALYZER products.

The above members are set up with print control statements to be printed by ICEGENER. Code the SYSUT1 and SYSUT2 DD statements as follows:

```
//SYSUT1 DD DSN=SYS90R.KCDBMS.INFO(JRNALYZ),DISP=SHR
//        DD DSN=SYS90R.KCDBMS.INFO(LOGANLYZ),DISP=SHR
//        DD DSN=SYS90R.KCDBMS.INFO(TASKNLYZ),DISP=SHR
//SYSUT2 DD SYSOUT=A,DCB=(LRECL=80,BLKSIZE=3120,RECFM=FBA)
```

CA-Datamacs

CA-Datamacs is a test data generator utility program. It works through simple COBOL like control statements embedded in either a complete or abbreviated COBOL program. Standard COBOL source statements supply much of the information about the files, while CA-Datamacs control statements specify which files are to be created, the number of test records to generate, and the contents of the data fields within the records.

Features

CA-Datamacs supports sequential, VSAM, and CA-IDMS files. Test data may be created either from scratch or from already existing files using the file processor capability of CA-Datamacs. A command level language, referred to as CA-Datamacs command language (DCL), provides the ability to match files, merge files, update a file in place, and convert files.

CA-Datamacs/CA-IDMS Interface

CA-Datamacs supports CA-IDMS integrated indexes and provides the facility for the following CA-IDMS tasks:

1. CREATE - Builds records as described by subschema record descriptions and altered by CA-Datamacs field modifiers, and stores them in a database.

2. UNLOAD - Writes selected database records to a sequential data set.

3. LOAD - Writes selected records, with any specified alterations, to a database.

4. MODIFY - Retrieves, alters, erases, connects and/or disconnects records to the database.

5. COPY - Builds records, as described by subschema record descriptions and altered by CA-Datamacs field modifiers, and writes them to a flat file.

Cataloged Procedures

Procedure	Description
KKDMXLV	Invokes CA-IDMS Local Version
KKDMXCV	Invokes CA-IDMS Central Version

JCL Examples

CA-IDMS UNLOAD task using local version -

```
//STEP01   EXEC KKDMXLV
//DM.STEPLIB DD
//         DD DSN=              (application loadlib)
//         DD DSN=              (CA-IDMS loadlib)
//DM.CARDIN  DD *
//         DD DSN=              (source CA-Datamacs PGM)
//DM.IDMSOUT DD DSN=            (file for unloaded database)
//DM.xxxxxx  DD DUMMY           (Replace xxxxxx with DDNAME of your
                                 journal file.  Must include one DD
                                 for each journal at your site).
//DM.yyyyyy  DD DSN=            (data dictionary)
                                (Replace yyyyyy with DDNAME of your
                                 CA-IDMS dictionary).
//DM.zzzzzz  DD DSN=            (PDS of DB area to be used)
                                (Replace zzzzzz with DDNAME of DB
                                  area.  Must include one DD for each
                                  DB area used in execution).
/*
```

CA-IDMS UNLOAD task using Central Version -

```
//STEP01   EXEC KKDMXCV
//DM.STEPLIB DD
//         DD DSN=              (application loadlib)
//         DD DSN=              (CA-IDMS loadlib)
//DM.SYSCTL  DD DSN=            (SYSCTL file)
```

```
//DM.CARDIN  DD *
//         DD DSN=                (source CA-Datamacs PGM)
//DM.IDMSOUT DD DSN=      (file for unloaded database)
 /*
```

CA-UNIPACK/Quality Assurance (QA)

The CA-UNIPACK/QA is available to NITC customers to provide support in standards, procedures and guidelines in the IDMS environment. The following products make up the CA-UNIPACK/QA package.

CA-IDMS/ENFORCER: The CA-IDMS/Enforcer provides automatic verification and enforcement of naming standards, entered into the IDMS Integrated Data Dictionary (IDD).

CA-IDMS/STANDARDS ADMINISTRATION SYSTEM ONLINE: The Standards Administration System Online (SASO) is an interactive online tool that enhances the communication of standards and guidelines. SASO provides alternative access, including text editing, to the printed Standards, Procedures, and Guidelines (SP&G) manual.

CA-IDMS/STANDARDS, PROCEDURES, AND GUIDELINES (SP&G): The Standards, Procedures, and Guidelines (SP&G) manual is a document provided to help establish standards, procedures, and guidelines for an IDMS environment. A basic version of the manual is provided at install time that may be tailored to the user's particular environment.

NOTE: Please note that only DBA types with CA-ACF2 read permissions to KCDBMS high-level qualifier data sets can install the CA-UNIPACK/QA products.

TRACER

Tracer is a flexible software tool, from Allen Systems Group, Inc., that reformats database record images from IDMS journal archive files, converting them from a cryptic state to the format that appeared to the original updating application program.

Tracer can also produce an audit trail of database updates compiled from an IDMS journal archive file as well as many other different types of available standard audit reports. For more detailed description of Tracer options see Chapter 1 of the Tracer User's Guide.

Tracer Installation

To make use of the Tracer product customers should include the following files in their Tracer JCL:

KSYS1.KCTRCR.APAR.LOADLIB
KSYS1.KCTRCR.LOADLIB

Both of the above loadlibs should be included in your TRACER JCL. The APAR.LOADLIB should always be concatenate first.

The DSN below contains the necessary sample JCL needed to begin using TRACER. Customers should create their own copy of this DSN for their own site-specific use.

KSYS1.KCTRCR.JCLLIB

The DSN below contains the necessary COBOL source to create site specific or standard Tracer audit reports. Customers should also create their own copy of this DSN for their own use.

KSYS1.KCTRCR.SRCLIB

Tracer References

To order manuals, send purchase orders to the following:

Tracer User's Guide

Allen Systems Group, Inc.
1333 Third Avenue, South
Naples, Florida 34102

CA-IDMS References

To order CA-IDMS manuals, write or call:

Computer Associates International
2291 Woodoak Drive
Herndon, VA 20171

Attn: Federal Division

(703) 709-4500

Addition reference information may also be obtained by accessing url: http://www.ca.com/ and entering in the SEARCH box: **CA-IDMS**.

Chapter 5. System 2000 (S2K) and Related Products

Introduction

System 2000 (S2K) is a Database Management System from the SAS Institute. S2K provides the user with the ability to define new databases, modify the definition of existing databases, and to store, retrieve, and update values in these databases. S2K allows the organizational structure of the database to represent information about the data. The designer of a S2K database determines what relationships are relevant and how data will be represented.

S2KCLIST s

The following CLIST s may be used for online access to S2K and S2K PLEX compilers. (You must be prepared to provide the DATABASE NAME (DBN), password, and data set names of your S2K database.)

S2K CLIST s	Description
%S2KN	Invoke the base S2K version 11.6
%S2KNTST	Invoke the base S2K version 12.0

Cataloged Procedures

Base S2K commands may be executed in batch, as well as PLEX (Programming Language Extension) procedures. PLEX is used for both retrieving data from and updating S2K databases via high-level programming languages.

The Center presently maintains a basic S2K procedure and PLEX procedures for MVS COBOL and VS Fortran. These are single-function PLEX procedures that may be used in combination:

SYSTEM 2000 Version 11.6

Procedure	Description
S2KN	Basic S2K execution 11.6
S2KFVSP	Execute PLEX VS Fortran pre-compile 11.6
S2KFVSC	PLEX VS Fortran compile output from the pre-compile. 11.6
S2KFVSL	PLEX VS Fortran link output from the compile. 11.6
1S2KMVCP	PLEX IBM Cobol execute S2K pre-compile. 11.6
S2KMVCC	PLEX IBM Cobol compile output from the pre-compiler. 11.6
S2KMVCL	PLEX IBM Cobol link output from the compile 11.6
S2KMVCG	PLEX IBM Cobol execute Procedural Language program. 11.6

SYSTEM 2000 Version 12.0

Procedure	Description
S2KNT	Basic S2K execution 12.0
S2KFVSPT	Execute PLEX VS Fortran pre-compile 12.0
S2KFVSCT	PLEX VS Fortran compile output from the pre-compile. 12.0
S2KFVSLT	PLEX VS Fortran link output from the compile. 12.0
S2KMVCPT	PLEX IBM Cobol execute S2K pre-compile. 12.0
S2KMVCCT	PLEX IBM Cobol compile output from the pre-compiler. 12.0
S2KMVCLT	PLEX IBM Cobol link output from the compile 12.0
S2KMVCGT	PLEX IBM Cobol execute Procedural Language program. 12.0

Basic S2K JCL -

```
//Job statement …
//STEP1   EXEC S2KN,S2KPARM=N15456,REGION=2000K
//EMPLOYE1  DD DSN=KCCS2K.S2K.EMPLOYE1,DISP=OLD
//EMPLOYE2  DD DSN=KCCS2K.S2K.EMPLOYE2,DISP=OLD
//EMPLOYE3  DD DSN=KCCS2K.S2K.EMPLOYE3,DISP=OLD
//EMPLOYE4  DD DSN=KCCS2K.S2K.EMPLOYE4,DISP=OLD
//EMPLOYE5  DD DSN=KCCS2k.S2K.EMPLOYE5,DISP=OLD
//EMPLOYE6  DD DSN=KCCS2K.S2K.EMPLOYE6,DISP=OLD
//EMPLOYE7  DD DSN=KCCS2K.S2K.EMPLOYE7,DISP=OLD
//EMPLOYE8  DD DSN=KCCS2K.S2K.EMPLOYE8,DISP=OLD
//SYSOUT DD SYSOUT=*
//SYSIN  DD *
USER,password:
DBN IS dbnname:
(Insert S2K commands here)
EXIT:
/*
//
```

PLEX IBM COBOL procedures in combination -

```
//Job statement …
//STEPPR   EXEC   S2KMVCP
//S2KCP.SYSIN DD DSN=your.cobol.source,DISP=SHR
//STEPCC   EXEC   S2KMVCC
//STEPLK   EXEC   S2KMVCL
//STEPGO   EXEC   S2KMVCG
//databas1  DD DSN=your.s2k.databas1,DISP=SHR
//databas2  DD DSN=your.s2k.databas2,DISP=SHR
//databas3  DD DSN=your.s2k.databas3,DISP=SHR
//databas4  DD DSN=your.s2k.databas4,DISP=SHR
//databas5  DD DSN=your.s2k.databas5,DISP=SHR
//databas6  DD DSN=your.s2k.databas6,DISP=SHR
//databas7  DD DSN=your.s2k.databas7,DISP=SHR       (optional)
//databas8  DD DSN=your.s2k.databas8,DISP=SHR       (optional)
```

```
//SYSIN    DD *
//SYSOUT   DD SYSOUT=*
```

GENIUS

Genius is a conversational facility that helps produce S2K reports. One or more reports can be specified in a single Genius session. Customers should be familiar with the basic concepts of S2K database definitions and ACCESS where-clauses and if-clauses. Genius is available for both 11.6 and 12.0 Versions of S2K databases.

Genius can be used by novice end-users or by experienced database users. Genius produces error-free REPORT language commands on a file that is read by S2K to produce the actual report(s).

All syntax used in the S2K CLIST is valid in the GENIUS CLIST .

File name GENDSC is required to contain a DESCRIBE of the database. If a DESCRIBE file does not already exist for the database, one must be created before GENIUS can be executed. To create a GENDSC file, type the following at ISPF option 6 or the READY prompt:

1. **ALLOC F(GENDSC) DA('kccs2k.s2k.employe.GENDSC.DATA') NEW CATALOG TRACKS SPACE(1,1) LRECL(80) BLKSIZE(6160)**

2. **%S2KN** (for a release 11.6 database) or
   %S2KNTST (for a release 12.0 database)

3. Respond to normal CLIST parameter questions:
   user,password:
   dbn is dbnname:
   report file is gendsc:
   describe:
   exit:

This will place the database definition (DESCRIBE output) into the data set allocated to GENDSC.

Genius is invoked from option 6 for the READY prompt by typing **%GENIUS** for release 11.6 databases or **%GENIUSTS** for release 12.0 databases.

References

For a Publications Catalog or to order manuals, write to:

SAS Institute, Inc.
SAS Campus Drive
Cary, N.C. 27513-8000

(919) 677-8000

Chapter 1. BookManager

Introduction

BookManager is an IBM software product designed to offer online books at your terminal or workstation. Online books are structured much like printed books. A book can have a title page, front matter sections such as a table of contents, abstract and preface, lists of figures and tables, a body consisting of text sections, and back matter sections such as a glossary and an index.

Online books have many qualities similar to printed books, but offer more flexibility in displaying, searching, and organizing information. An entire library or a specific book can be browsed or selected book parts can be accessed using BookManager's search capabilities.

BookManager READ electronically remembers and recalls previous topics seen, saving the time that would be required to browse hardcopy publications in an attempt to locate previously viewed information. Revisions to a document may also be reviewed without having to read through the entire publication. Generating hardcopy reports of selected topics are also available.

Usage

BookManager may be accessed from TSO or the NITC Web Server.

Web Access

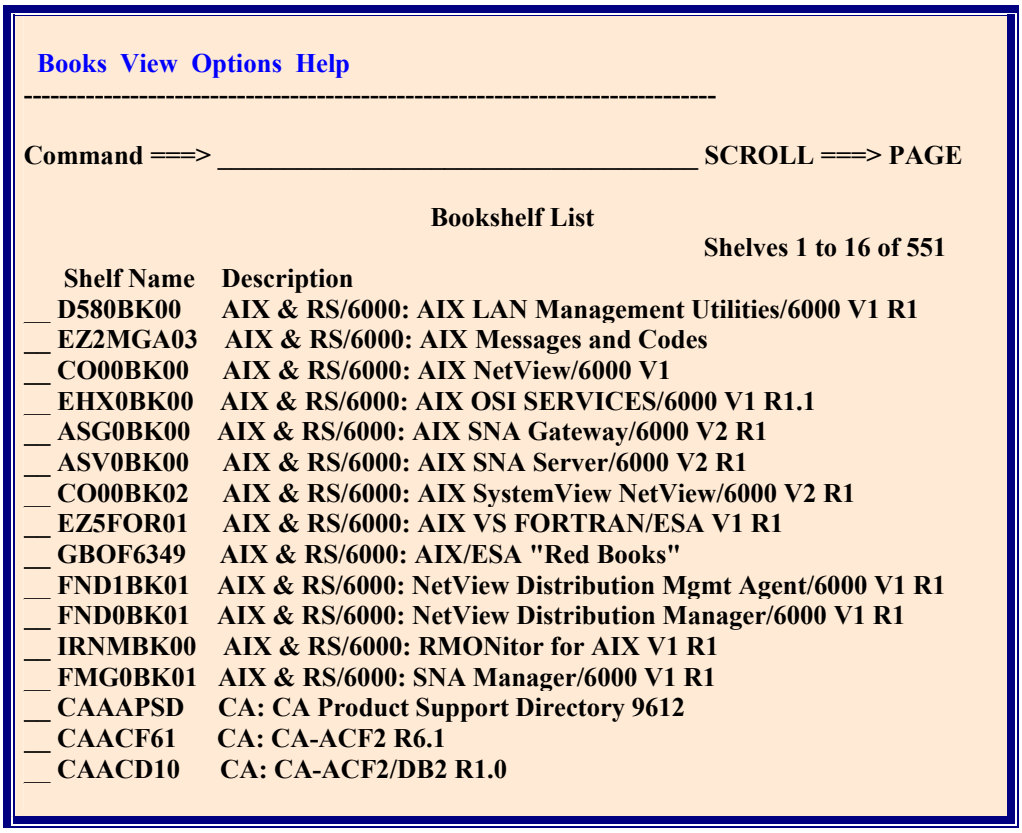
Go to www.agbiz.usda.gov and click on BookManager. Then click on BookManager Format Books. The Bookshelves page displays.

Notice that the available bookshelves are listed alphabetically by title in the left column with their associated shelf names in the right column.

You can scroll the bookshelf list or find a specific bookshelf by keying the title or shelf name in the entry box.

TSO Access

Sign on to TSO and select ISPF Option **=D.BM**.



Each panel has an action bar, status line, scrollable area, command line, and scroll field. You tell BookManager READ what actions to perform by making choices or typing information on a panel.

Action Bar

To select an action, use the Tab or cursor movement keys to position the cursor on your choice, then press ENTER. Once you select an action on the action bar, BookManager READ displays a pull-down with choices related to the action you selected.

Command Line

If you are an experienced user, the command line may be used to bypass the pull-downs and windows and use BookManager READ, ISPF, and Time Sharing Option (TSO) commands to complete certain actions.

Status Line

A status line is displayed on the right side of a panel or window beneath the title. The status line indicates the number of items or lines currently displayed in the list or topic, and the total size of and your current location in that list or topic.

Scrollable Area

Beneath the action bar of each panel is a scrollable area that contains either a list or text. In a bookshelf list you see a list of bookshelves; in a bookshelf you see a list of books, and in a book you see text. You scroll backward or forward to see additional information. If the scrollable area is wider than the screen, you can scroll left and right to see additional information.

In the scrollable area of a bookshelf list or bookshelf, you can select the bookshelves or books you want to work with before you select actions you want to perform.

To mark items for selection in a scrollable area, type a slash (/) over the underscore in front of the bookshelf name, book name, or listed choice. When you select an action bar item, BookManager READ performs the action for the marked items only.

Function Keys

To see a description of the function key settings on any current panel:

1. Select Help on the action bar.
2. Select Keys help in the Help pull-down.

Search for Specific Text

With BookManager READ, you can do far more than just read a book online. You can search for words and phrases in a single book, or in all the books on a bookshelf. BookManager READ shows you a list of the topics, or a list of the books, containing information that matches your search request.

The following is an example of a search in BookManager:

- Place a slash (/) by the bookshelf to be searched
- Place a slash (/) by the book to be searched (or press <ENTER> to search the entire bookshelf)
- Press PF10 (to take the cursor to the action bar at the top of the screen)
- Select Option 2 from the search window
- Type the search criteria in the window that appears.
- The system will display the books/topics that a match is found for the selected search criteria.
- Place the cursor on the book or topic you want to view.
- PF6 (Find the next best topic on search criteria)
- PF7 (Sequential read Page Up)
- PF8 (Sequential read Page Down)
- PF3 (Exit)

HELP

You can press **F1** to get help at any time. The following BookManager Help is available:

1. If a message is displayed, you will see an explanation of the message.

2. If the cursor is on an input field, you will see contextual information for the field.

3. If the cursor is on an action bar item, you will see information about the action.

4. If the cursor is on a choice in a pull-down or window, you will see information about that specific choice.

5. If the cursor is in a window but not on a choice, you see information about the actions available in the window.

6. If the cursor is anywhere else on the panel, you see help for the panel.

7. If the cursor is on a reference phrase (at the end of some helps), you see information about the subject of that phrase.

You can also get help while viewing a panel by selecting Help on the action bar.

Close BookShelf

To close the currently displayed bookshelf:

1. Select Books on the action bar.

2. Select Close bookshelf in the Books pull-down. BookManager READ displays the Close Bookshelf confirmation window.

3. Select one of the following:

   Yes, to close the bookshelf and return to the bookshelf list, bookshelf, or book from which you requested it. If you opened the bookshelf or listed books directly from the TSO/E READY mode message or ISPF menu, the session ends and you return to TSO or the ISPF menu.

   No, or press F12 to remove the Close Bookshelf confirmation window

Printing

While looking at a book online, you can print the current topic or a range of topics. Select Services, option 2 (Print), and select the print topics. You will see the following Hardcopy Panel. Modify the print options as desired:

```
                        Hardcopy Utility                Enter required field
Command ===>

Process option     1. Print and keep data set or member
                   2. Print and delete sequential data sets

  Data Set Name  . . 'usesrid.KCCBKM.ENU.EOX03M07.PRINT'
  Volume Serial  . . . .          (If not cataloged)


Print Mode . . . . . . . BATCH       (BATCH or LOCAL)

Sysout class . . . . . . A              (BATCH only)
Local printer ID or
writer-name  . . . . . . U999            (LOCAL only)

Job statement information: (If not to local printer/external writer, verify
before proceeding)
===> //xxxxxxx JOB (xxxxxxxxxxxxxx,xxxx),'your name',
===> //            CLASS=K,MSGCLASS=R,MSGLEVEL(1,1),
===> //            PRTY=11,TIME=1
===> /*ROUTE PRINT vpsprinter#
```

References

The BookShelf name 'EZ2MVEEA' contains the reference manuals for detailed information on using BookManager or select Help on the Action Bar.

Chapter 2. Phoenix (CBT)

Introduction

The NITC provides USDA customers with the PHOENIX Presentation System for Computer Based Training (CBT). The only requirement to register for CBT training is customers have a TSO user ID.

CBT Information

Information on available courses, how to register, and how to take a course may be found in Teleview NEWS. To access this information, you must sign on to Teleview. (For information on connecting to Teleview, see the Teleview chapter of this Handbook.) Once in Teleview, type **NEWS** at the command prompt. This menu will display:

```
NATIONAL INFORMATION TECHNOLOGY CENTER

    1 - NITC Internet Website Info
    2 - Computer Based Training (CBT)
    3 - NITC Operational Schedule
    4 - Exit From This MENU

    Enter Option ---->
```

Select option 2, Computer Based Training (CBT):

```
    Computer Based Training (CBT)

  1 - Sorted Course Lists
  2 - Course Descriptions
  3 - How to Register
  4 - How to Sign On
  5 - Exit From This MENU

  Enter Option ---->
```

1- Sorted Course Lists - This option will provide a complete list of all courses available. You will have the option to select a list sorted by Course Name or by Course Number.

2 – Course Descriptions - This option provides a submenu of course topics:

```
AVAILABLE COURSE TOPICS

 1 - Introduction to Data Processing
 2 - Business and Technical Writing
 3 - MVS Courses
 4 - Application Programming
 5 - Database Management Systems (DBMS)
 6 - SAS
 7 - CICS
 8 - Client/Server Technology
 9 - VAX/VMS
10 - VM/CMS
11 - CA-7 / CA-11
12 - Networks / Data Communications / VTAM
13 - Project Management
14 - Miscellaneous Software
15 - Exit From This MENU

Enter Option ---->
```

Selecting one of these options will present options listing what courses are available for each of these topics. By selecting a specific course, the following is an example of the information presented about the course:

```
Course ID:   ISPFPD
Course Name: ISPF/PDF: Productivity and Programming

 Prerequisites:  Basic knowledge of ISPF functions and features.

 Objectives:
 The purpose of this course is to show you how to -
  1)  Fully exploit familiar ISPF commands and features.
  2)  Identify useful ISPF commands and features that you may have
      forgotten or never fully understood.
  3)  Expand your working ISPF tool set with techniques which may be
      new since your initial training.

 Estimated time:  8 hr 25 min

 To print a workbook for this course:
     Browse dataset KCUSER.PHOENIX.CNTL(PRNTWKBK) and (SAMPLE)
```

This information will help you decide what course will provide you with the training you want, how long the course will take and how to obtain a workbook (if one is available).

How to Register

Once you have decided which course(s) you would like to take, contact your agency training coordinator. Please refer to the course number when requesting a course.

Your training coordinator will contact NITC to get you registered for the course. If you are unsure who the coordinator is at your agency, contact your agency Helpdesk.

Requests for enrollment should be emailed to teresa.hurley@usda.gov. Please include the course id of the course(s) you wish to enroll in and your agency, name and phone number. You will receive a reply that will notify you that you have been enrolled in the course(s) you requested and provide you with the student id you have been assigned.

Taking a Course

To begin taking the course:

1. Sign on to **TSOB**
2. Go to option 6 or the READY prompt
3. Type: **KCCPHNX**
4. Enter the Student Id and Course Number. (No password is required)

To exit the course, type **SIGNOFF** where the cursor is positioned or follow the instructions given at the beginning of the course.

Chapter 3. Statistical Analysis System (SAS)

Introduction

The SAS system consists of software products that enable you to access, manage, analyze, and present all of your data. With the SAS System you can perform data entry, retrieval, and management; report writing and graphics design; statistical and mathematical analysis; business forecasting and decision development; operations research and project management; and applications development.

The core of the SAS system is Base SAS software. It consists of:

- The SAS Language – A programming language used to manage your data.
- SAS Procedures – Software tools for data analysis and reporting.
- A Macro Facility – A tool for customizing programs and reducing text.
- The DATA Step Debugger – Tool to find logic problems in DATA step programs.
- The SAS Windowing Environment – Interactive graphical interface used to run and test SAS programs.
- The Output Delivery System (ODS) – System that allows the customer to route output based on destinations selected through the use of ODS statements.

SAS Language

The SAS language consists of statements, expressions, functions, options, informats, and formats. You can program any number of analyses and reports from it. The SAS system also has a library of built-in programs known as SAS Procedures. These procedures use data supplied by you to produce preprogrammed reports, requiring minimal effort from you.

SAS Programming

A SAS program is made up of a sequence of statements executed in order. A statement is used to provide information or instructions to SAS and must be appropriately placed in the program. A SAS program typically consists of two steps:

1. The DATA step begins with a DATA statement and can include any number of program statements. The keyword DATA is followed by the name that is to be used to define the SAS data set.

   Data steps are used to read and modify data; print reports; write out disks or tape; read external raw data files; and produce new SAS data sets by subsetting, merging, and updating the old data sets.

2. The PROC step begins with a PROC statement. The keyword PROC is followed by the name of a procedure such as PRINT, SORT, or COPY. SAS software provides a variety of procedures that are used to produce reports, compute statistics and perform utility operations.

DATA and PROC steps can appear in any order, and any number of each can be used in a SAS program. When you submit a DATA step to the SAS system for execution, it is first compiled and then executed.

SAS Output

SAS output is the result of executing SAS programs. Most SAS procedures and some DATA step applications produce output. Base SAS software produces two forms of printed output:

1. The SAS Log, which can include a listing of your program statements, system options specified, any warning or error messages, how many variables and observations each data set contains, how much time each step required and a listing of data lines

2. The results from most SAS procedures and some DATA step applications

SAS Statements

As with any language, there are rules to follow when writing SAS programs. The rules are as follows:

- SAS statements must end with a semicolon.
- SAS statements can be entered in lowercase, uppercase, or a combination of the two.
- Any number of SAS statements can appear on a single line.
- A SAS statement can be continued from one line to the next, as long as no word is split.
- Words in SAS statements are separated by blanks or by special characters (such as the equal sign and the minus when used in a calculation).

SAS Names

- SAS names are used for SAS data set names, variable names, and other items.
- Most SAS names can be up to 32 characters in length.
- The first character must be a letter or an underscore (_).
- Subsequent characters must be letters, numbers, or underscores.
- Blanks cannot appear in SAS names.

SAS Data Sets

The SAS system organizes data into a rectangular form called a SAS data set. A SAS data set, also called a table, consists of observations and variables. Each row represents an observation and each column represents a variable.

Every SAS data set consists of a descriptor portion, which describes the variables in the data set; and a set of observations, which contain values for the variables. If the descriptor and the observations are in the same location, the data set is called a SAS DATA FILE. If the descriptor and the observations are stored separately, they form a SAS DATA VIEW.

A collection of SAS data files is called a SAS Data Library.

SAS Data Library Architecture

SAS Data Library DCB attributes:

DSORG=PS
RECFM=FS
BLKSIZE=4096 to 27648 (Must be in increments of 512)
LRECL=BLKSIZE (The LRECL must be equal to the value specified in the BLKSIZE attribute)

It is recommended that you use the largest optimal BLKSIZE (27648) if your library is likely to contain members with large numbers of observations. The default is 6144.

SAS Procedures

SAS software consists of components called procedures that allow you to perform many different kinds of analysis and data management functions, as well as produce many different types of text-based and graphical presentation output.

SAS software provides a variety of procedures that are used to produce reports, compute statistics, and perform utility operations:

1. Report procedures are used to display useful information, such as data listings (detail reports), summary reports, calendars, letters, labels, forms, multipanel reports, and graphical reports.

2. Statistical procedures are used to compute elementary statistical measures which include descriptive statistics based on moments, quantiles, confidence intervals, frequency counts, cross-tabulations, correlations, and distribution tests. They also rank and standardize data.

3. Utility procedures are used to perform basic utility operations. They create, edit, sort, and transpose data sets, create and restore transport data sets, create user defined formats, and provide basic file maintenance such as to copy, append, and compare data sets.

MACRO Facility

The Macro facility is a tool for customizing your SAS programs and for reducing the amount of text required to perform common tasks. The macro facility allows you to establish large or small groups of common SAS programming statements and assign a name to those groups of text. This reduces the size of your program because you to refer to the assigned name of the group of programming statements in your program rather than the text itself.

The macro facility consists of two components:

1. The macro processor
2. The macro language

The macro processor performs the work and the macro language is the syntax used to communicate with the macro processor.

The following delimiters trigger macro processor activity:

```
&name
%name
```

&Name refers to a macro variable and is called a macro variable reference. %Name refers to a macro. The text substitution produced by the macro processor is completed before the program text is compiled and executed.

Defining a Macro Variable

A macro statement must begin with a %MACRO statement, contain a name for the macro, and end with the %MEND statement.

Example:

```
%MACRO CREATE;
      DATA TEST;
       SET INFLE.MASTER;
       RUN;
%MEND CREATE;
```

DATA Step Debugger

The Data Step Debugger is an interactive programming tool that allows you to find logic problems within your DATA step program. The Data Step Debugger allows you to issue commands to execute DATA step statements one by one and then pause to display the resulting variables' values in a window.

By observing the results that are displayed, you can then determine where the logic error lies. This process can be repeated as many times as needed in a single session. To

invoke the debugger, add the DEBUG option to the DATA statement and execute the program.

SAS Windowing Environment

The SAS windowing environment is an interactive environment used for editing, testing, debugging, and prototyping programs and applications.

The following windows are available under the SAS Windowing Environment:

1. The Explorer window allows you to view and manage SAS files and creates shortcuts to non-SAS files. The SAS Explorer window is invoked by issuing a PROC build statement or the EXPLORER command from within the SAS windowing environment.

2. The Results window allows you to navigate and manage output from SAS programs that you submit.

3. The Program Editor, Log, and Output windows allow you to enter, edit, and submit SAS programs; view messages about your session and programs that you submit; and browse output from the programs you submit.

4. Other windows that are not a part of base SAS software can be accessed through and used in display manager include SAS/FSP windows such as FSBROWSE, FSEDIT, FSLETTER, FSLIST, FSPRINT, and FSVIEW. Other examples include SAS/GRAPH windows such as GRAPH, LEGEND, PATTERN, and SYMBOL.

Output Delivery System (ODS)

The Output Delivery System (ODS) allows printed output more flexibility. ODS can produce output data sets from procedure output, customize reports created by procedures, and send output to a variety of destinations. ODS supports the following destinations:

1. HTML destination, which produces output that is formatted in Hypertext Markup Language.

2. The LISTING destination, which produces output that is formatted like traditional SAS Procedure output.

3. The PRINTER destination, which produces output that is formatted for high-resolution printers.

4. The OUTPUT destination, which produces SAS data sets.

ODS destinations can be open or closed. When the destination is open, ODS can send output objects to it. When the destination is closed, ODS cannot send output objects to it. An open destination uses resources even if no output is directed to it. To conserve

resources, do not leave a destination open if it is not needed. ODS statements are used to open and close the appropriate destination.

Other SAS Products

The following products are available to NITC customers in addition to Base SAS Software:

- SAS GRAPH
- SAS/ASSIST
- SAS/SHARE
- SAS/CONNECT
- SAS/ACCESS (Interface to DB2 and ORACLE Software only)
- SAS/AF Software
- SAS/FSP
- SAS/ETS
- SAS/IML
- SAS/STAT

SAS/GRAPH

SAS/GRAPH software enables you to generate output that graphs the relationship of data values to one another, to enhance existing graphs, or to simply create graphics output that is not tied to data. SAS/GRAPH software can produce various kinds of graphics output:

- Charts
- Plots
- Maps
- Text
- Three-dimensional graphs

SAS/GRAPH software consists of procedures that produce graphics output by using data from SAS data sets. This graphics output is made up of commands that tell a graphics device how to draw graphics. Graphics output can be:

- Displayed as graphics output on your device, optionally, using the display manager GRAPH windows.
- Written as a file of graphics commands
- Written to a SAS catalog entry for later replay
- Printed on a graphics hardcopy device

SAS/GRAPH software also includes a SAS data library, which contains SAS/GRAPH map data sets. These data sets can be used to produce geographic maps. To access the SAS Maps library, you must use the SAS system option MAPS='SYS90R.SAS.MAPS' in your configuration file, at SAS invocation, or in an option statement.

SAS/ASSIST

SAS/ASSIST software is a menu-driven, task-oriented interface to the SAS system. It can be used to access the SAS system without having to learn SAS programming statements. SAS/ASSIST builds and documents SAS programs based on the tasks selected from SAS/ASSIST online menu.

Note: You must have a graphics terminal to access any graphics or icons within SAS/ASSIST.

The SAS/ASSIST Primary menu consists of the following tasks:

- TUTORIAL: Provides information on how to use SAS/ASSIST and the SAS system.
- DATA MANAGEMENT: Allows you to enter new data, look at and change existing data, import data from an external file, access data from external databases, and copy, subset and sort data.
- REPORT WRITING: Allows you to create or modify a variety of reports.
- GRAPHICS: Allows you to produce high and low resolution graphics including vertical, horizontal, stacked, and grouped bar charts; pie charts; line plots; and maps of the world.
- DATA ANALYSIS: Provides several analysis tools.
- PLANNING TOOLS: Enables you to design and analyze experiments, produce control charts, perform project management tasks, and forecast time series.
- EIS: Allows you to build menu-driven applications for executive information systems. (Not Available)
- RESULTS: Allows you to process any of the programs that SAS/ASSIST creates and documents. You can recall, delete, or rename a program or output; edit or execute a program; or redisplay a graph.
- SETUP: Allows you to associate reference names with any data libraries or files you need to use, manage device drivers, printing forms, sorting data, and reviewing your current key settings.
- EXIT: Allows you to exit SAS/ASSIST software.

SAS/SHARE

SAS/SHARE software is for applications that require concurrent update access to SAS files. You can perform the following when users have concurrent update access to SAS files:

While one user is creating a member in a data library, others can create, read, delete, and update members in the same library.

While one user opens the ACCESS or DIR window on a data library, other users can open the same window to browse, delete, edit, or rename members in the same library.

While one user opens the CATALOG window on a catalog, other users can open the CATALOG window to browse, copy, delete, or rename entries in the same catalog.

While one user is in the process of editing a SAS data set, other users can update the SAS data set, read the SAS data set as input, add observations to the SAS data set, and copy the SAS data set that is being updated as well as other members of the library to another library.

SAS/CONNECT

SAS/CONNECT gives you access to the files, hardware resources, and SAS software on various remote hosts to use with a SAS session on the local host. With SAS/CONNECT, you can:

1. Develop and test programs in the local SAS session and then remote submit the same program to the remote host to process large data sets stored on the remote host.

2. Implement full macro processing on both the local and remote hosts. You can remote-submit macro steps to the remote host and then pass return code information about the remote process to the local SAS session.

3. Build SAS applications that perform tasks in multiple environments. A single SAS program can contain statements that run locally and statements that execute on multiple remote hosts.

4. Upload or download SAS data sets, SAS catalogs, and external files in either text or binary format.

5. Execute graphics programs on the remote host and display the graphics locally using the graphics capability of the local workstation, plotter, or printer.

SAS/ACCESS

SAS/ACCESS software provides an interface between the SAS system and data in other software vendors' database management systems (DBMS). NITC is currently licensed for the DB2 and ORACLE interface. These interfaces consists of the following procedures and interface engine:

- The DBLOAD procedure
- The Interface View engine
- The ACCESS procedure

- The SQL Procedure Pass-Through Facility
- The SAS Explorer window
- The SAS/ACCESS LIBNAME statement for Relational Database Interfaces

The DBLOAD procedure is used to load SAS or other data into a DBMS table and to submit dynamic non-query DBMS SQL statements to a DBMS database.

The Interface View engine allows you to read and write data directly to and from files formatted by a database management system (DBMS), such as DB2 and ORACLE. Interface view engines enable you to use SAS procedures and program statements to process data values stored in these files without converting and storing them in files formatted by SAS.

The ACCESS procedure is used to create and edit access descriptors and view descriptors, and to create SAS data files. Descriptor files describe DBMS data so that you can read, update, or extract the DBMS data directly from within a SAS session or in a SAS program

The SQL Procedure Pass-Through Facility is used to send DBMS-specific SQL statements directly to a DBMS for execution. The Pass-Through Facility uses a SAS/ACCESS interface engine to connect to the DBMS.

The SAS Explorer window replaces the ACCESS Window and enables you to navigate within an interactive SAS session by using a new, point-and-click interface.

The SAS/ACCESS LIBNAME is used to assign a SAS libref directly to a relational DBMS (RDBMS) or to objects in an RDBMS, such as tables and views. You no longer need to create access and view descriptors, although all previously created access and view descriptors are fully supported in Version 7 and later.

SAS/AF

SAS/AF software allows you to develop interactive applications within the SAS system. SAS/AF provides an interface to all the data access, management, analysis, and presentation features of the SAS System.

SAS/AF applications are stored in SAS catalog entries. The BUILD procedure (or the BUILD command) is used to create the following types of catalog entries:

FRAME entries, which consist of object-oriented components used for developing applications in environments that support a graphical user interface

PROGRAM and MENU entries, which provide character-based features for developing applications in environments where a graphical user interface is not available

SCL entries, which store SAS Component Language (SCL) programs

CBT and HELP entries provide information and assistance to application users.

SAS/FSP

SAS/FSP procedures provide interactive facilities for data entry, editing, and retrieval. With SAS/FSP software you can:

1. Browse and edit the contents of SAS data sets
2. Enter data into existing SAS data sets
3. Create new SAS data sets
4. Browse and edit SAS data views created with SAS/ACCESS software
5. Browse SAS data views created with the SQL procedure in Base SAS software
6. Create, edit, and print form letters and reports
7. Build and customize end-user applications.

SAS/ETS

SAS/ETS software provides tools for a wide variety of applications in business, government, and academia. Major uses of SAS/ETS procedures are economic analysis, forecasting, economic and financial modeling, time series analysis, financial reporting, and manipulation of time series data.

SAS/IML

SAS/IML (Interactive Matrix Language) SAS/IML is powerful software that is used to create matrices. You can access built-in operators and call routines to perform complex tasks such as matrix inversion or eigenvector generation. You can define your own functions and subroutines using SAS/IML modules. You can operate on a single value or take advantage of matrix operators to perform operations on an entire data matrix.

SAS/IML software provides many ways to create matrices. You can create matrices by doing any of the following:

- Entering data yourself as a matrix literal
- Using assignment statements
- Using matrix-generating functions
- Creating submatrices from existing matrices with subscripts using SAS data sets

SAS/STAT

SAS/STAT consists of procedures that are used to provide basic statistical analyses and associated graphics. You can compute descriptive statistics, perform simple hypothesis

tests, fit statistical models with regression and analysis of variance, and perform survival analysis as well as some multivariate analyses.

SAS Access

The SAS system may be accessed online (windowing environment mode) or in batch (background mode).

Windowing Environment Mode

Enter **KKSAS** at the READY prompt or option 6 of the ISPF Primary Option panel or you may also select option or **=D.K** of the ISPF Primary Option Panel.

Select 2 to access the SAS Program Editor.

To exit online SAS, type **BYE** on the command line of the Program Editor or any other window.

Batch Mode

Cataloged Procedures

Procedure	Description
KKSAS	SAS batch procedure

Execute the cataloged procedure KKSAS in your batch JCL.

(To access DB2 or ORACLE, you must use the LOAD option to concatenate the desired Database load library with the SAS load library.)

Sample SAS JCL:

```
//jobstatement…
//STEP1  EXEC KKSAS,LOAD='DB2.LOAD.LIBRARY'
SAS program code
```

Training

Instructor based training courses are provided by SAS Institute. You can contact SAS Institute at 1-919-531-7321 for more information on courses available in your area. Course listings are also available through the SAS website at www.sas.com/training

Online training is also available via TSO. See the Computer Based Training chapter of this Handbook for more information on available CBT training.

References

Online SAS documentation may be viewed through the NITC website at www.agbiz.usda.gov/sas8doc/onldoc.htm or www.agbiz.usda.gov/sas8doc/main.htm.

A complete listing of all available reference titles are available through the SAS website at www.sas.com/pubs. Reference materials may be ordered by contacting SAS Institute at 1-800-727-3228.

Chapter 4. PILOT (CCPLUS) EIS

Introduction

Executive Information System (EIS) is an electronic means of providing managers and executives with access to information that will help them manage their time and resources most effectively.

Command Center

Command Center is a development environment that provides all the tools needed to create and modify application-specific Executive Information Systems. It distributes processing between the mainframe and your personal computer. On the mainframe, the system performs most number crunching and all the data retrieval, information-sharing, and maintenance functions. The primary function of the personal computer is to implement the menus and graphics commands sent from the mainframe. The personal computer also performs all the calculations required to scale, create, and place graphs, menus, figures, and text.

The Command Center consists of six basic subsystems:

- The Language Processor
- The DataBridge Import and Export facility
- The Database
- The Menu Control System
- The Graphics module
- The Communications Processor

Language Processor

The Core of the Command Center's power and versatility is its Command Center Language. Command Center Language is a fourth-generation language (PLI) designed to facilitate the creation of Executive Information Systems.

The Command language consists of two types of action-producing verbs called commands and functions. Commands are instructions to the system to perform a task. Functions also request performance of a task, but they always return information, which may be data or a status indicator.

Frequently Used Commands and Functions:

ASK() - Function that returns a response. Example - name = ASK("What is your name?")

CALL - Command that executes the specified procedure. Example - CALL loadste(ts1,ts2,ts3,ts4)

CALL MENU - Command that executes the specified menu, returning to the calling procedure or menu.

CLS - Command that clears the screen: also restores full screen functionality after using graphic mode.

COMPILE - Command that compiles the specified procedure. Example - COMPILE loadste

COPY - Command that copies an existing file or files. Example - COPY loadste oloadste

DATE - function that returns the current system date. Example - PRINT DATE()

EDIT - Command that opens a file for editing, creating it if necessary, using your system-defined editor. Example - EDIT proc loadste

ERASE - Command that removes an existing file from the system. Example - ERASE errors

EXIT - Command that ends the Command Center Session and returns to the host operating system. Example - EXIT noflush

LIST CATALOG - Command that displays a list of files in your Command Center catalog. Example - LIST CATA

MESSAGE - Command that prints text on the 25th line of the screen. Example - MESSAGE red "Hello"

PATH - Command that establishes that catalog(s) to be searched during execution. Example - PATH pltapp

PCVERSION() - Function that returns the version number of the workstation program. Example - PRINT PCVERSION()

PRINT - Command that displays the result of a function or the value of variables and arithmetic operations. Example - PRINT tot

RENAME - Command that changes a file's name. Example - RENAME loadste loadst

RUN MENU - Command that executes a menu located in the specified menu file. Example - RUN MENU ad$mnu:top

SET SYSHELP PILOTHELP - Activates the on-line system help file.

SHOW PATH - Command that displays catgroup(s) in the current path.

TIME - Function that returns the current system time.

TYPE - Command that displays the contents of a text-readable file to the screen.

VERSION - Command that displays the Command Center host version number.

To make applications run as efficiently as possible, the language processor also includes a compiler. The compiler parses and validates all language statements so that the run-time environment can operate as efficiently as possible.

Databridge Import and Export Facility

The DataBridge facility serves as a link between the Command Center and the outside world, it allows you to capture, extract, and format data to suit the needs of your application. The Data Import System allows you to capture files from non-Command Center sources and perform some fundamental data extraction.

Database

Command Center is not a Database PER SE. It is actually a keyword-driven data storage and manipulation system designed to offer three major features:

- Manipulation of time-series data commonly used by executives
- Optimization for fast retrieval, rather than updating, because most users read, rather than write, data
- "Normalization" of data from different sources, so that all are items comparable

The data storage facility consists of one or more tables, each of which contains multiple records. Each record consists of a series of fields. Thus, a table consists of a set of common fields that are used to describe the features of distinct items.

The system allows you to define four types of fields:

- Character fields may contain only character string data (up to 255 characters, including blanks); this is the default field type.
- Numeric fields hold numeric variables of Real data type.
- Timeseries fields may hold only time-series data (up to 366 points).
- Key fields are used to facilitate the retrieval of records.

Optimal Sizes

While there is no one set size of a table, record, or field, the following range of parameter values is suggested for optimal performance:

Optimal Table and Record Sizes –

Item	Size
Table	May contain up to 10,000 records
Record	May contain up to 450 fields (each record requires 50 bytes of overhead)

Field Sizes-

Type	Overhead	Size
Character	2 bytes	1 byte per character
Timeseries	116 bytes	8 bytes per time-series point
Numeric	8 bytes	No additional

Create Data Table

To create a data table, you simply name the table in a "MAKE TABLE" command. You then define the fields in the table via the "MAKE FIELD" command.

Once you have defined the table and the fields, load the data into the table. There are two ways to do this:

The "MAKE RECORD" command allows you to name all fields and values for them.

Or

Build the record field-by-field by issuing a series of "BUILD RECORD" commands, each of which specifies a different field, followed by a single "MAKE RECORD" command.

Example:

1. The following command creates a table called SALES:

   MAKE TABLE sales

2. Once the table has been created, define the fields in the table, using "MAKE FIELD" commands. Each command contains the name of the field and the type of field it is. The default is character.

   MAKE FIELD product
   MAKE FIELD agent NUMERIC
   MAKE FIELD prior TIMESERIES
   MAKE FIELD budget TIMESERIES
   MAKE FIELD actual TIMESERIES

3. After the table and its fields exist, you can create the records by using the "MAKE RECORD" or "BUILD RECORD" command.

   MAKE RECORD FIELD product "CARS", FIELD agent 120, -
   FIELD prior TSLIST(5,10,20,9) START 1/89 QUARTERLY, -
   FIELD budget TSLIST(8,12,15,9) START 1/89 QUARTERLY, -
   FIELD actual TSLIST(9,11,12,8) START 1/89 QUARTERLY, -

Menu Control System

The Menu System is a Command Center subsystem that consists of three component parts:

1. The Menu Editor allows you to create and modify the menus with which you build applications.

2. The Menu Commands and Functions allow you to operate on menus from within menu buttons and procedures.

3. The Run Time Processor handles the execution of all menus and the commands they contain.

The Menu System runs on both the PC and the mainframe. Menu commands run on the mainframe, then display their results on the PC screen. In addition, the mainframe controls the contents of menus and the actions taken when menu buttons are selected.

Graphics Module

This module consists of a comprehensive set of commands used to control the display of data in graphic form, taking advantage of the personal computer's graphics capabilities.

The Graphics Module interacts with the data manipulation facility to retrieve requested information and sends formatting commands and data to the PC. The

Graphics system offers both a charting and a drawing capability. The charting system allows you to easily create a wide variety of charts. The drawing system allows you to draw geometric figures and text.

Communications Processor

The Command Center Communications System consists of one module on the PC and one on the mainframe. These two modules allow the mainframe and the Personal Computer to talk to each other using a common interface. The Communications System provides 3270 terminal emulation for coaxial IBM communications.

Installing PC Software

To install the personal computer software, perform the following steps:

1. Request a copy of the installation diskette from the Pilot system administrator.
2. Follow instructions on the installation diskette.

Entering PILOT Command Center

To get into Command Center, you must start the personal computer portion of the software, connect to the mainframe, log into the mainframe operating system, then start the mainframe version of Command Center. Once you reach the mainframe, enter your TSO logon id and password.

At the "READY" prompt enter **CCPLUS** and press **<ENTER>**. You should receive a "?" prompt. You are now in CCPLUS.

Build ADVANTAGE/G Application

1. Enter PATH PLTAPP at the "?" prompt. This will point you to the PILOT system files.

2. Enter RUN GN:TOP at the "?" prompt.

3. Select Advantage/G to proceed to the Advantage Generation Menu.

4. Select the PREFIX utility to create an Advantage/G instance. To create an instance, you must choose a unique two-character prefix that will identify the Advantage/G instance that you are building. The first time that you run Generation, the prefix AD automatically appears. This is a sample instance provided with Advantage/G.

5. Enter the instance name and title.

6. Select the Database Definition and specify the data table that your Advantage/G instance is to query.

7. Select the Keys/Period Definition to define the keys that identify the entry point in the data table, the periodicity of the data, the fiscal start period, and whether the data is to be displayed in YTD or current format.

8. Select the Import Definition to specify the type of import:

9. Advantage/G or user-written procedure.

10. Select the Data Column Definition to specify the position and contents of columns on standard reports.

11. Select the Label Column Definition to specify the position of labels on the Standard Report and line item coloration based on exception formulas.

12. Select the Exceptions Definition to specify exception formulas, trigger types, and trigger settings.

13. You can select the Help Menus if you wish to include explanatory text for Standard and Exception Reports.

14. Select Standard Report Chart Editor to select data to be presented, presented, colors of the graph components, and charts behind data columns on the Standard Reports.

15. Select Exception Detail Report Chart Editor to select colors of graph components and charts behind data columns on the Exception Detail Reports.

16. Select Bottom Buttons if you wish to change the functionality behind the buttons at the bottom of the Standard Report.

17. Select Generate to create the new Advantage/G instance as specified by the definitions you have selected above.

18. Select Exit to return to the Generation Menu.

19. Select Exit to return to the "?" prompt.

Loading ADVANTAGE/G Application

User written procedures consisting of Import commands and functions are used to capture and extract data from non-Command Center files. A procedure is a collection of system statements treated as a unit within a file.

A single procedure may occupy an entire file, or you may place several procedures within the same file. For the latter, procedures must be enclosed by "PROCEDURE" and "PEND" statements. To create a procedure, issue the command "EDIT ROCEDURE" and supply a procedure name.

EDIT PROCEDURE procname

Procname is the name of your procedure. You will be in TSO edit and may type your Import commands or functions at this time. PF3 will save your data and take you back to the READY PROMPT. Enter PILOT to get back into the PILOT system. All procedures must be compiled before execution. To compile your procedure type the following at the "?" mark:

COMPILE Procname

To execute a compiled procedure type the following at the "?" mark:

CALL Procname

Testing ADVANTAGE/G Application

1. Type PATH PLTAPP at the "?" prompt. This will point you to the PILOT system files.

2. Type RUN ADXXMNU:MAINT (where XX represents the unique instance prefix you created above)

Running ADVANTAGE/G Application

1. Enter PATH PLTAPP at the "?" prompt. This will point you to the PILOT system files.

2. Enter RUN ADXXMNU:TOP (XX represents the unique instance prefix you created above)

Exiting Command Center

1. Use the RETRACE or RETURN buttons to back out of current screens until you reach the main menu. Press the EXIT button to return to the CCPLUS system. You should be back at the "?" mark. Enter EXIT NOFLUSH to leave the CCPLUS system.

2. You should be at the TSO READY prompt. IF not enter =X to exit any ISPF screen you may be in. Enter LOGOFF to leave TSO.

User Requirements

Contact the CCPLUS System Administrator to request a copy of the CCPLUS installation diskette.

Contact the CCPLUS System Administrator to register you to the CCPLUS Account Registry. This will allow your account to access the CCPLUS software. Your account is usually your logon id. Files created by CCPLUS will be prefixed with your account.

If you wish to access other accounts within your agency, contact your Security Officer for ACF2 permissions. You will need to use the "PATH Account" statement before accessing an application with an account other than your own.

Select TSOA when accessing the mainframe. CCPLUS software can only be accessed through the "TSOA" system.

CCPLUS software is written in PLI and may not be supported in future releases of the IBM OS/390 operating system. There are no plans for ACCRUE software to provide any enhancements or fixes to this software in the near future.

Each Agency is responsible for developing and maintaining their CCPLUS applications.

References

Fundamentals of Command Center
Command Center Dictionary Volume I: A through M
Command Center Dictionary Volume II: N through Z
Personal Computer Operations Guide
System Administrator's Guide for MVS/ESA
Enter-View User's Guide
Worksheet, Retrace, and Electronic Paperclip User's Guide
Advantage/G User's Guide
Advantage/G Implementation Guide
SnapShot User's Guide
Worksheet User's Guide

Manuals may be ordered by contacting:

ACCRUE Software
40 Broad Street
Boston, Ma 02109

Phone (617) 374-9500

Additional reference materials may be obtained from url: http://www.accrue.com/

Chapter 5. Preference

Introduction

Preference is a Computer-Based Reference (CBR) system that allows you to create, store, and retrieve online information. Reference material stored in Preference can be presented in several different forms:

- Online manuals, guides, references
- Online help
- Printed documentation

Online manuals, guides and references: Entire online references can be created with Preference. The Reader and Writer guides supplied with Preference are examples of a complete online reference.

Online help: Preference can be used to create help for an application. Help may be based on your screen page, your cursor location, or the data you entered as input. Online help can be either full-reference information or pop-up help windows.

Printed documentation: Reference material stored in Preference can be printed. The printed version can be a full manual, complete with a table of contents, glossary, and index.

Preference Users

Preference provides three user access levels. Each level has its own range of activity and functions. Preference permits each level to access any level below it in the hierarchy. The Three levels are listed below:

- The Supervisor
- The Writer
- The Reader

The Supervisor: Person responsible for operating and maintaining the Preference system. The Preference administrator is also the supervisor.

The Writer: Person who uses Preference to organize and store information. A writer creates and updates the reference material in Preference.

The Reader: Person who accesses Preference to obtain information written and stored in Preference. A reader can control the amount of information retrieved and the sequence in which the information is viewed.

Preference Volumes

Preference stores information in an online volume. The volume is organized as follows:

- A volume contains 1 to 26 chapters, labeled A to Z.
- Each chapter contains topics, numbered from 1 to 32.
- Each topic contains 1 to 16 sections, labeled A to P.
- Each section contains up 99 reference items. Each item is a screen or window of online information.

Most of the information in a volume exists in the items that make up the chapters, topics, and sections. However, a writer can also create:

- A prefix
- Chapter abstracts
- A glossary
- An index

Accessing a Volume

Readers can access a volume in one of three ways. You can provide standalone, concurrent, or programmable access to a volume.

Stand-alone Access: Access is gained through the sign-on screen. The readers have access to all the information on the volume.

Concurrent Access: Readers with concurrent access to a volume can request help from an application screen. Requesting help can move the reader directly into a volume, where they can access all parts of the volume. A pop-up help window can provide a menu that also allows a reader to move into a volume. Once in the volume, readers can access all parts of the volume.

Programmable Access: The application determines when to provide help and what help is available. Help can be entering a volume, where readers can access all parts of the volume. Or help can be a pop-up window that allows the reader to move into a volume.

Communication Environment

Preference can be set up to run under TSO or VTAM and has an interface to IDMS, ORACLE, or CICS. Access is restricted to those regions that run on TSOA.

Requirements

Each Agency is responsible for assigning a Preference administrator to establish and maintain that agency's Preference regions.

NITC will provide assistance to the Agency Administrators on the initial setup and installation of Preference software as needed.

Each Agency's administrator is also considered the Preference Supervisor and is responsible for registering all Writers and Readers to that Agency's Preference volumes.

Each Agency is responsible for the maintenance and backup of that Agency's Preference system files.

Preference regions may only be executed on the TSOA system due to license restrictions.

References

Preference Writer Guide Release 7.7.0
Preference Reader Guide Release 7.7.0
Preference Writer Guide to Concurrency Release 7.7.0
Preference Supervisor Guide 7.7.0
Preference Messages and Codes Guide 7.7.0
Preference System Guide 7.7.0

Manuals may be ordered from PATHLORE:

PATHLORE
Corporate Headquarters
7965 North High Street, Suite 300
Columbus, OH 43235

Phone: 1-888 – 728-4567

Additional reference information may also be obtained by accessing url: http://www.pathlore.com/ and entering in the SEARCH box: **PREFERENCE**.

Training

Online training is available for Preference:

Preference/Writer (GSWRIX)
Preference/Reader (GSREAX)
Preference/Concurrency (GSCONX)

See the Computer Based Training chapter of this Handbook for more information on accessing this and other online training.

Chapter 6. CA-Easytrieve Plus

Introduction

CA-Easytrieve Plus is an information retrieval and data management system designed to simplify computer programming. Its English-like language and simple declarative statements provide the new customer with the tools to produce comprehensive reports with ease, while its enhanced facilities provide the experienced data processor with the capabilities to perform complex programming tasks. It supports sequential, VSAM, IAM and CA-IDMS data sets.

Cataloged Procedures

Procedure	Description
KEZTP	CA-Easytrieve Plus
KEZTPIDD	CA-Easytrieve Plus w/ CA-IDMS & IDD interface
KEZTPIAM	CA-Easytrieve Plus for IAM data sets

Example JCL

```
//STEP01  EXEC KEZTP
//SYSIN   DD  *
//       DD  DSN=    ... (CA-Easytrieve Plus source PGM)
//  (Any additional input or output data set DD's)
```

CA-Easytrieve Plus with CA-IDMS interface -

```
//STEP01  EXEC KEZTP
//   LOAD1=  ... (PDS of application loadlib in quotes)
//   LOAD2=  ... (PDS of CA-IDMS loadlib in quotes)
//EZTP.aaaaaa  DD DUMMY  (aaaaaa is DDNAME of your journal data sets)
//EZTP.bbbbbb  DD DSN= ... (PDS of data dictionary and bbbbbb is DDNAME of
                                your CA-IDMS dictionary)
//EZTP.cccccc  DD DSN=    ... (PDS of DB areas to be used replace cccccc with
DDNAME of
                                your DB area)
```

CA-Easytrieve Plus with CA-IDMS and IDD interface -

```
//STEP01   EXEC KEZTPIDD
//   LOAD1=  ... (PDS of application loadlib in quotes)
```

```
//   LOAD2=  ... (PDS of CA-IDMS loadlib in quotes)
//EZTPIDD.SYSIN DD  *
//            DD  DSN= ...  (CA-Easytrieve Plus source PGM)
//EZTPIDD.xxxxxx DD DUMMY  (xxxxxx is DDNAME of your journal data sets)
//EZTPIDD.yyyyyy DD DSN=... (PDS of data dictionary and yyyyyy is DDNAME of
                                    your CA-IDMS dictionary)
//EZTP.aaaaaa   DD DUMMY    (aaaaaa is DDNAME of your journal data sets)
//EZTP.bbbbbb   DD DSN=... (PDS of data dictionary and bbbbbb is DDNAME of
                                   your CA-IDMS dictionary)
//EZTP.cccccc   DD DSN=... (PDS of DB areas to be used, replace cccccc with
DDNAME
                                   of your DB area)
```

CA-Easytrieve Plus using IAM data sets -

```
//STEP01   EXEC KEZTPIAM
//IAM.IAMCNTL DD  *
  PGM=EZTPA00
  ISAM=DATA SETA,IAM=DATA SETAA
//IAM.SYSIN  DD  DSN= ...(CA-Easytrieve Plus source program)
//IAM.FILEA  DD  DCB= ...(Input model data set)
//IAM.FILEAA DD  DSN= ...(Input data set)
```

Interactive Access (TSO/E)

There are several CA-Easytrieve Plus CLIST s available that may be executed from ISPF option 6 or the READY prompt. The first eight CLIST s invoke foreground execution of CA-Easytrieve Plus, and the last two submit jobs to the background reader for batch execution.

%EASYP01 has five required positional parameters:

1. DSNAME of the data set containing the CA-Easytrieve Plus statements.
2. DSNAME of the data set containing the input data.
3. Project account number.
4. A job name
5. The user LOGONID.

There are also eight optional key word parameters.

%EASYP02 can be used to enter short, simple CA-Easytrieve Plus jobs, where the output is printed back to the terminal, and selected records are written to an output data set. The CA-Easytrieve Plus statements are entered through the CLIST . There are four required positional parameters:

1. The DSNAME of the input data set.
2. The DDNAME of the input data set.
3. The DSNAME of the output data set.
4. The DDNAME of the output data set. (This must be an existing data set)

%EASYP03 is used to execute an CA-Easytrieve Plus job, print out is received at the User's terminal, and direct output to FILEB (not saved). There are three required positional parameters:

1. The DSNAME of the data set containing the CA-Easytrieve Plus statements (SYSIN).
2. The DSNAME of the input data set.
3. The DDNAME of the input data set.

%EASYP04 is identical to %EASYP03, except that data sets C through F are not allocated.

%EASYP05 is identical to %EASYP02 except that the third positional parameter is not present; thus, output data going to FILEB cannot be saved.

%EASYP06 is similar to %EASYP02, but has several differences. There is no output saved in FILEB. There are two positional parameters:

1. The DSNAME of a data set containing CA-Easytrieve Plus library statements and user statements (LIBRARY). Additional statements may be added online. When the message 'INPUT' is received at the terminal, user statements may be entered.
2. The DSNAME of the input data set (DATAIN).
3. The DDNAME of the input data set (INDDN). Specifying LIST will cause the records contained in the LIBRARY data set to be listed at the terminal.

%EASYP07 is identical to %EASYP06, except that a third and fourth positional parameter is added. The DSNAME of an output data set, and the output data set DDNAME.

%EASYP08 is the same as %EASYP04, except that third and fourth required positional parameters are used, as in %EASYP07.

%EASYP09 submits an CA-Easytrieve Plus job for batch execution. It accomplishes the same purpose, as % EASYP01 except the print data set cannot be received at the terminal. Any cataloged data set can be used for input, including those residing on tape. Output data set cannot be saved. There are six required positional parameters that are identical to those in %EASYP01. There are also five optional key word parameters.

%EASYP10 is the same as %EASYP09, except that the first positional parameter is replaced by two positional parameters, each of which is the DSNAME of a data set containing CA-Easytrieve Plus statements. Two job steps are included in the submitted job. The first creates an output data set, under control of the first set of CA-Easytrieve Plus statements. The output is then passed to the second step for additional processing under control of the second set of CA-Easytrieve Plus statements.

References

Online documentation for CA-Easytrieve Plus is available using IBM's BookManager. For help accessing or using BookManager, see the BookManager chapter of this handbook

CA-Easytrieve Plus reference manuals may be ordered by contacting Computer Associates:

Computer Associates International
2291 Woodoak Drive
Herndon, VA 20171
Attn: Federal Division

(703) 709-4500

Addition reference information may also be obtained by accessing url: http://www.ca.com/ and entering in the SEARCH box: **CA-Easytrieve**.

Chapter 7. CA-Easytrieve Key

Introduction

CA-Easytrieve Key (EZKEY) is a menu-driven, full-screen, context-sensitive editor that can be used to develop CA-EASYTRIEVE PLUS programs. It is a complete system for creating and maintaining CA-EASYTRIEVE PLUS programs.

EZKEY may be used regardless of your level of data processing knowledge or programming expertise. Two input modes (PROMPT and FREE) are designed to accommodate all levels of data processing experience. You can SWAP back and forth between the two modes as often as you like.

Features

EZ/editor is a full-screen, SPF-like editor with a complete set of profile, primary, and line commands. Profile variables that control the appearance of your data may be changed at any time during a session. These values are saved between edit sessions in your profile.

Primary commands allow you to move around within a member being edited and to copy or move data from another member.

Line commands enable you to insert, delete, copy, move, shift, overlay, repeat, and exclude individual lines or blocks of lines within a member. You can define a line tag.

Commands issued during an edit session are displayed at the left of the top line of your screen. Short error messages are displayed at the right of this line.

EZ/checker is a syntax checker for the CA-EASYTRIEVE PLUS language. EZ/checker interfaces with EZ/editor upon your request. The specific services you desire and the time they are performed is controlled by the CHECK primary edit command and the setting of the profile variables ERRORS and CHECK.

If you request EZ/checker services during an edit session, each new or changed data line is checked for syntax and semantic errors each time you press ENTER or any PF key. A message is inserted under every error detected. This checking service is also performed for any CA-EASYTRIEVE PLUS program that is copied or moved into the member being edited.

Access

CA-Easytrieve Key may be invoked by typing **%EZKEY** at ISPF option 6 or the READY prompt.

EZKey Menu	Description
PF Keys	Display Current Program Function Key Settings
EZ/EDIT	Create or Modify an CA-EASYTRIEVE PLUS Program
EDIT	Create or Modify any Member
UTILITIES	Perform EZ/Key Library Maintenance
FOREGROUND	Run an CA-EASYTRIEVE PLUS Program On-Line
BATCH	Submit a Program for Batch Processing
EZ/REPORT	Report Processing Facility
TUTORIAL	EZ/KEY Tutorial
EXIT	Exit EZKEY

Training

A complete EZKEY online tutorial is available within EZ/KEY. To begin the tutorial, invoke EZKEY and type **T** at the EZKEY primary selection panel.

References

CA-Easytrieve Key reference manuals may be ordered from Computer Associates:

Computer Associates International
2291 Woodoak Drive
Herndon, VA 20171

Attn: Federal Division

(703) 709-4500

Chapter 8. Computer Emergency Notification System (CENS)

Introduction

The National Information Technology Center (NITC) has implemented the Computer Emergency Notification System (CENS) application to enhance existing procedures for emergency planning and notification. The CENS application utilizes the internet/intranet to notify personnel of emergencies, building related alerts, and messages.

This application is a 2003 Federation of Government Information Processing Councils (FGIPC) Intergovernmental Solutions Awards Winner. It is being utilized at USDA Departmental Administration's Office of Operations, Veterans Administration Central Office and is being considered for immediate implementation at other government entities.

Benefits

CENS ensures that its user organizations are able to effectively and efficiently communicate with staff in the event of a crisis situation. The application requires no additional software other than the standard Windows environment.

CENS will - Deliver consistent messages to all personnel.
- Indicate specific problems and actions to take.
- Work continuously if the user is logged on to the system.

Key Features

- Effectively and efficiently communicates with staff.
- Capable of reaching the greatest number of people in affected areas.
- Able to reach people inside closed dwellings with high ambient noise levels.
- Able to reach those in peripheral areas of a workplace.
- Authorized personnel perform CENS administration and use.
- Complies with Section 508 of the Rehabilitation Act.
- Functions in the Microsoft environment.
- Minimal impact to network performance.
- Utilizes security features.
- Is flexible and adaptable.
- Low cost solution.

Processes

The CENS system environment consists of three separate processes:

Message Generator - Authorized personnel create appropriate emergency messages for distribution by the CENS server. CENS has integrated security controls that restrict message-sending capability to specific workstations using authorized access controls.

Server - A centralized CENS server provides message storage using a Microsoft SQL Server database. Immediate distribution of information, via the web, is provided to each person's workstation running the CENS client.

Client on Individual Workstation - The CENS client software resides on individual workstations. Using standard HTTP protocols, the CENS client automatically queries the CENS server at configured intervals to determine if a new message is waiting. When activated, the server sends the message to the CENS client. The CENS client takes over the workstation, displays a message with a flashing background, and emits a sound.

For emergency messages, a siren sound precedes the message. You will hear a beep sound for alerts and information messages.

The CENS application must be actively acknowledged and closed by the end user before any other use can be made of the workstation.

NITC CENS Services

The CENS application is available to NITC customers upon request. NITC will provide any combination of the following services for a fee:

- Acquiring and/or hosting a CENS server or utilize customer owned Windows server.
- Technical project planning and coordination.
- Applications development expertise to modify the most current USDA code and customize the CENS application to meet customer requirements.
- CENS application level support during the development phase and future.
- Coordination with customer emergency command personnel to create a database of message types and content.
- Technical assistance with systems setup, installation, screen development, deployment, testing, and technical support.
- Application level Help Desk.
- Application level knowledge database available for testing and deployment.

- Systems and end user CENS documentation.

References

The CENS application is available upon request. An online presentation explaining this application is available for your review at <http://www.ocio.usda.gov/nitc/products/index.html>.

Please contact me at 888-USE-NITC or send an email to NITCCENS@usda.gov for information on how to activate CENS at your site or request additional information.