H623 USER-Access®

for IBM AIX Systems

Release 5.0.1 M12 for AIX 4.3

Release 5.0.1 M13 for AIX 5.2

Memo To Users

April 15, 2005

6420 Sycamore Lane N #300
Maple Grove, MN 55369

Introduction

This product implements the Network Executive Software, Inc. (NESi) H623 USER-Access for IBM eServer pSeries AIX systems. The remainder of this document will refer to these systems as AIX. H623 can run as a NetEx/IP or TCP/IP application.

USER-Access file transfers support large files. Please read the Service Notes (below) for important information regarding large file support.

This product has been tested on AIX 4.3 and AIX 5.2.

This product uses the *Hxx3 (Rel 2.0) USER-Access for UNIX User Guide*. Refer to "Documentation Updates", which describes features that are not documented in the User Guides.

New Features

There are no new features.

Service Notes

The following are known problems and/or limitations with this release of AIX USER-Access:

- USER-Access file transfers will support large files on versions of AIX that provide this feature (e.g., 4.2). If USER-Access is installed on an OS version prior to 4.2, it will be configured without large file support. If, after installing USER-Access on such a system, the system is upgraded to a version of AIX that supports large files (e.g., 4.2), then you must re-install USER-Access before large files can be transferred to or from the AIX system. No warning will be issued. Large file transfers without large file support will simply fail after the 2 GB limit has been reached.
- This product supports only the SYSTEM default method of user login authentication as described in the system file /etc/security/user.
- When running LOCAL or REMOTE commands, if there are commands within an invoked shell script file that you do not want to execute when under USER-Access, you can test for an environment variable USERA, defined to be "yes" by USER-Access. For example, if you enter the USER-Access command "remote myscript", and "myscript" is a Bourne shell script containing an "echo Hello World" command, you can avoid the "Hello World" output during USER-Access LOCAL or REMOTE invocation by making the following edit:

  ```
  if [ "$USERA" != "yes" ]
  then
  echo Hello World
  fi
  ```

  The USERA environment variable is always defined under USER-Access as described above and can be tested for within any shell script file.
- If three keyboard interrupts in a row are pressed in response to an ASK -SECURE command, the user's terminal characteristics may not be reset (i.e., the terminal may be left in no echo mode). In this case, the user should reset the terminal manually with an 'stty echo' command.

Installation Notes

This release of USER-Access runs as either a NetEx/IP or TCP/IP application. At installation time the user must determine on which type of protocol USER-Access is to run. If there are installation differences for NetEx/IP or TCP/IP they are noted by preceding the pertinent section with the line "For Network_type:", where "Network_type" is either NetEx/IP or TCP/IP.

Prerequisites

- Access to the root account.
- Up to 12 Mbytes of disk space. This can be divided over two file systems, if necessary, with up to 8 Mbytes for a DISTRIBUTION directory, and up to 4 Mbytes for a TARGET directory.
- The install script will ask for two UNIX UserID names, one as the owner of all installed files, and one (optionally) as the UserID used to perform CAM backups. The owner UserID must exist at installation time. The CAM backup UserID does not necessarily have to exist at installation time, as it is simply written to a file for validation during backups.
- The installation script requires the use of the make and ld (linker) utilities. These utilities are released with the AIX Operating System. If they are not installed on your system, you will need to install the "Application Developer's Toolkit" (ADT) from your AIX installation media.

For NetEx/IP:

- H620IP NetEx/IP must be fully installed, tested and running.

For TCP/IP:

- The host must be a functioning node on a TCP/IP Local Area Network.

Installation Procedure

Before proceeding with the installation, make certain that you have all of the items listed in "Prerequisites" at the beginning of this document. For the purposes of this document, we will assume the installation is being performed on an AIX 5.2 system.

If this is an update to USER-Access, be sure to stop the old Service Initiator. The Service Initiator can be stopped using the provided utility '/usr/nsc/sicom/si/control'. Replace the hostname 'localhost' with the name of your local host (for more details, see "Stopping the Service Initiator" in this document):

```
$ /usr/nsc/sicom/si/control -h localhost stop
```

Installation and verification of USER-Access consists of the following steps, described in detail on the following pages:

1. Create the USER-Access DISTRIBUTION directory
2. Load the distribution
3. Install the distribution
4. Update network databases
5. Start the Service Initiator
6. Verify USER-Access
7. Set up automatic Service Initiator startup

Step 1. Create the USER-Access DISTRIBUTION Directory

Note: Skip this step and continue with "Step 2" if this release is being installed from CDROM.

The USER-Access release is loaded into a DISTRIBUTION directory and is then installed into the TARGET directory. Existing files in the TARGET directory are replaced with the new release files. Any user-modified control files are automatically preserved.

Nothing is modified in the DISTRIBUTION directory during the installation process allowing repeated installations into different TARGET directories, each with different install options. To preserve disk space the DISTRIBUTION directory can be deleted once USER-Access is fully installed into the TARGET directory.

The installation procedures and examples assume a DISTRIBUTION directory of /usr/nsc/ua_m13 and a TARGET directory of /usr/nsc/sicom. However, any valid directory (with adequate space) can be selected. If alternate directories are chosen, be sure to replace /usr/nsc/ua_m12 and /usr/nsc/sicom with the alternate directories in the commands that follow.

At this point, you should create your DISTRIBUTION directory and set your current default directory:

```
$ mkdir /usr/nsc        (if it does not exist)
$ chmod 755 /usr/nsc
$ mkdir /usr/nsc/ua_m13
$ cd /usr/nsc/ua_m13
```

Step 2. Load the Distribution

The USER-Access release for AIX is distributed as a single TAR file on CDROM, magnetic tape, or a single TAR file on diskette.

For CDROM:

Mount the CDROM to a named file system entry point (e.g. /cdrom). The specific mount command is determined by the system software components installed on your UNIX system. The CDROM format is HSFS (High Sierra) compliant and requires that the destination system be able to process this format. Refer to your UNIX Operating System manual or UNIX System Administrator for assistance in mounting the CDROM.

Extract the distribution from the TAR file:

```
$ tar -xvf /cdrom/H623-m12.tar
```

For Magnetic Tape:

To load USER-Access distributed as a single TAR file on magnetic tape, use the following procedure, replacing '/dev/rst/0' below with the name of your local tape device. Load the distribution into your current default directory:

```
$ tar -xvf /dev/rst/0
```

Note: For best results, load the distribution directly from the distributed media to the node on which the files will be stored. If this is not possible, your current node must be allowed root access to the file server node on which the files are loaded.

Step 3. Install the Distribution

The USER-Access installation script prompts for a valid install option, the TARGET directory, and the type of network (NETEX, TCPIP or USERGATE). Hit ENTER to take the default response shown within square brackets.

If this is an update installation (i.e., an earlier version of USER-Access already resides in the selected TARGET directory), then existing executables will be overwritten, but the prior version of each updated text file (e.g., a script or configuration file) will be preserved with "_save" appended to the name.

(**For TCP/IP:** For a TCPIP installation, the script also prompts for CAM Client installation options.)

To run the install script, you will need to be logged in as "root". If you are not already logged in as "root", invoke /bin/su before proceeding. Invoke the install script to perform the installation procedures (a TCPIP installation is shown, and the optional CAMOBSI module installation is shown):

```
$ ./install
Building for AIX 5.x

The USER-Access install options are:
ALL       - full USER-Access install
CAM       - install CAM Motif graphical client only
CART      - install CART utility only

Enter install option [ALL]?

A single Unix UserID is assigned ownership of all installed
USER-Access files and directories.  The choice of UserID may
be based on accounting practices or maintenance
responsibilities.  This UserID must already exist.

Enter the USER-Access UserID [root]?
```

```
The TARGET root directory defines the location where
USER-Access is installed.  This directory and subordinates
will be created if necessary.

Enter the target directory [/usr/nsc/sicom]?

USER-Access network choices are:

    TCPIP     - native TCP/IP connection
    NETEX     - direct Netex connection
    USERGATE  - TCP/IP connection to a Netex gateway

Enter the desired network [TCPIP]?

The Motif CAM client allows CAM access using the Motif
graphical user interface.  It is not required for the
USER-Access installation.

Would you like to install the Motif CAM client [yes]?


A statically linked Motif CAM client is provided on this
release.  However, you may wish to link the Motif CAM client
with your local Motif libraries.  Re-linking can potentially
provide an executable that uses a later version of these
libraries.  Also, if shared libraries are provided on your
system, re-linking will generate a smaller executable, and
may provide other performance benefits.

If you choose to re-link, you will be prompted for the
location of your X and Motif libraries.

Re-link the Motif CAM client with your libraries [yes]?

Enter the full pathnames of the directories containing your X
and Motif libraries.  You may enter multiple directories,
separated by spaces.  You do not need to add /usr/lib.

X/Motif library locations []?

Note: The X and Motif libraries necessary for re-linking the
CAM client have been located on your system.

For easy access to the USER-Access executables, the
following symbolic links will be created in a bin directory
of your choosing:
    user     -> /usr/nsc/sicom/user/start_client
    uaserver -> /usr/nsc/sicom/user/start_server
    cam      -> /usr/nsc/sicom/cam/start_client
If general access to the USER-Access utilities is desired,
then these symbolic links should be installed in a common
bin directory.

If a naming conflict occurs, and the original is also a
symbolic link, the original will be replaced.  If the
original is not a symbolic link, it will be preserved as
<name>_save in the original directory.
```

```
Where should the symbolic links be located [/usr/bin]?

If the CAM UABACKUP or UARAW utilities will be used to back
up the system, and a Unix UserID other than root will be
used to perform those backups, then a privileged USER-Access
Responder must be installed and configured to allow the
selected UserID to have full access to the system volumes.

If you request that a CAM Backup UserID be assigned, you
will be prompted for a Unix UserID name.  The Responder
executable will be given root SETUID permissions, and a text
file named '.uabackup' will be placed under the root
directory.  This file will contain the name of the backup
UserID that you assign.

Do you require the assignment of a backup UserID [yes]?

The file '/.uabackup' will be created for you.  If the file
already exists, and the contents are different, the original
will be renamed '/.uabackup_save'.

CAM Backup UserID []?

This release contains a CAMOBSI executable which
can serve as a DataTools OBSI for backing up certain
databases via CAM.  If you have DataTools SQL-BackTrack
installed, you may wish to install CAMOBSI.

Install CAMOBSI [no]? yes
Where are DataTools products installed [/usr/datatools]?

Checking for type of DataTools SQL-BackTrack installation...

--- Creating directory /usr/datatools/obsi.cam/bin/obsi

--- Adding links to /usr/datatools/obsi.cam
    in the following directories:
/usr/datatools/sbacktrack-3.0.1/links

This release contains a CAMXBSA shared library which
can serve as a storage manager for backing up INFORMIX
databases via ONBAR and CAM.  If you have INFORMIX
installed, you may wish to install CAMXBSA.

Install camxbsa.a [no]? yes
What name should be used for the library [/usr/lib/ibsad001.a]?
=======================================
Starting USER-Access installation
  TARGET = /usr/nsc/sicom
  OWNER = root
  NETWORK = TCPIP
  BIN = /usr/bin
=======================================
      .
      .
      .
```

```
=======================================
USER-Access installation completed
=======================================
```

Between the 'Starting USER-Access installation' and the 'USER-Access installation completed' messages the following steps are performed:

- Create the USER-Access TARGET directories, if necessary.
- Bind the USER-Access modules: CLIENT, SERVER, CART, CONTROL, SVCINIT.
- Change the ownership of the Service Initiator to root and set the SETUID permission bit for this executable. This will allow the Service Initiator to run with root privilege, giving it permission to update login-accounting files (utmp and wtmp files).
- Copy the USER-Access control files to the TARGET directory. Any previous control files that were modified are preserved.
- Create two symbolic links, user and uaserver, that reference the installed USER-Access CLIENT and SERVER executables. The symbolic links will be created in the /usr/bin directory unless an alternate directory is specified. This allows all users with the selected directory in their search paths to run these executables. If symbolic links named "user" and "uaserver" already exist, they will be replaced. If a file or directory exists with one of these names, the original will be preserved with the name "user_save" or "uaserver_save", respectively.
- If requested, bind the CAM graphical client, copy the related control files, and establish a symbolic link named cam for the CAM client.
- If USERGATE is selected, bind USER-Gate modules and copy the USER-Gate control files. The symbolic links that were created will reference the USER-Gate executables.
- If requested, set up a privileged Responder by creating a text file named "/.uabackup", changing the ownership of the Responder to root, and setting the SETUID permission bit. See Documentation Updates for further information.
- If requested, copy the CAMOBSI module to the TARGET directory and create a symbolic link in the appropriate directory.
- If requested, copy the CAMXBSA library and ONBAR_CREATE module to the TARGET directory and create a symbolic link with the specified name.

Be sure to scan the generated output looking for any errors or warning messages.

Step 4. Update Network Databases

For NETEX:

No action is required at this point for NetEx.

For TCP/IP:

No action is required at this point for TCPIP.

For USER Gate (Satellite):

The following network control files must be updated to include USER Gate specific entries. Important note: If NIS is active in your network, these changes should be made on the master NIS server node.

In the /etc/hosts file, add a 'usergate' alias to the host entry of the gateway node. If the current entry for the gateway node 'gemini' was:

```
192.9.1.20      gemini
```

it would be changed to:

```
192.9.1.20      gemini usergate
```

For a native TCP/IP network the default TCP/IP port number 6900 is used by USER-Access. For a USER Gate installation the default TCP/IP port numbers of 6930 and 6940 are used by USER-Access. These port numbers are recommended for use by USER-Access. However, if other applications require these port numbers, see "Appendix A: Updating the TCP/IP network control files" for instructions to change these defaults.

Step 5. Start the Service Initiator

Important note: Before starting the Service Initiator, you may have to edit its startup script. If your system has been altered, for example, so that the UNIX login procedure uses non-standard prompts for "login: " or "Password:", or if the "Login incorrect" message has been changed, then the appropriate keywords will need to be likewise modified in the file /usr/nsc/sicom/si/startup. See "Service Initiator Keywords" later in this document for a description of each keyword.

For an overview of the Service Initiator program, see "Understanding the Service Initiator" later in this document.

To start the Service Initiator, issue the following command:

```
$ /usr/nsc/sicom/si/startup
```

The Service Initiator will log messages to the file /usr/nsc/sicom/si/svcinit.log. Use the following tail command to look at the log file:

```
$ tail /usr/nsc/sicom/si/svcinit.log
```

One of the messages should read:

```
USER  Service offered
```

This confirms that the USER-Access service 'USER' is being offered by the Service Initiator.

Step 6. Verify USER-Access

The USER-Access Initiator (client) is normally invoked with the command 'user'. If 'user' is already a command on your system (for some other purpose), 'usera' is recommended. The installation procedure automatically creates a symbolic link named 'user' in a selected 'bin' directory. If a file already had that name, it was moved to 'user_save'.

A simple verification script has been supplied as part of the USER-Access release. This verification script prompts you for the local host name of your system, and a valid username and password on this host (for verification purposes, the username and password specified should at least have the privileges

you do, specify your username and password to be safe). It attempts to connect back to the host, login, and send and receive several files. You also have the option of verifying the operation of Central Archiving.

To invoke the verification procedure, set your working directory to the 'user' subdirectory and invoke USER-Access with the input script 'verify.ua':

```
$ cd /usr/nsc/sicom/user
$ user verify.ua
```

You should have several messages printed to your terminal followed by a final "Verification Successful" message. If an error is encountered during the verification, the procedure will terminate and an error message will be printed.

Step 7. Verify CAM Client

If the CAM graphical client was installed, you may wish to run it to verify that the Motif interface is set up correctly:

```
$ cam &
```

You can also perform the following simple steps to verify basic functionality and connectivity:

1. Click on Cancel in the Connect window to dismiss the Connect window. Then click on Help and then Contents to verify access to the CAM help file. Finally, click on Close to close the Help window.
2. If a CAM Central Server is available on your network, connect to the CAM Central Server by clicking on Connect in the File menu and entering the server's TCP host name, your CAM user ID and password, and any other connect parameters required by your site. A successful connection will display a group of primary action buttons. This verifies basic communications.
3. Click on Exit in the File menu or click on the Exit button (if connected to a Central Server) to stop the CAM graphical client.

Step 8. Set Up Automatic Service Initiator Startup

The USER-Access Service Initiator can be started automatically at boot time by inserting several lines into your system startup file. This installation step is optional.

To have the Service Initiator start automatically at boot time, edit either /etc/rc.tcpip (for TCP/IP or USER-Gate installations) or /etc/rc.hypd (for NETEX installations) and add the following lines at the end of the file (Remember to replace /usr/nsc/sicom below with your definition of this root directory):

```
#  USER-Access Service Initiator startup
start  /usr/nsc/sicom/si/startup ""
```

At system boot time, the Service Initiator will be started with the process name SVCINIT. Its messages will be logged to the file /usr/nsc/sicom/si/svcinit.log.

At this point, you may wish to verify your changes by rebooting the system.

This completes the formal installation of USER-Access on your local host. However, it is important to continue on through this document.

Understanding the Service Initiator

Introduction to the Service Initiator

The Service Initiator (SI) is a program that services USER-Access CONNECT or LOGIN requests from hosts on the network. It runs as a detached process under the name SVCINIT. Upon seeing a network login request, SI allocates a pseudo tty pair, forks a child and executes the /bin/login utility. Login information is passed between SVCINIT and /bin/login via the pseudo tty pair. Upon a successful login, /bin/login executes the user's default shell. Under the shell, a USER-Access server (Responder) process is then invoked. It is this server process that ultimately communicates with the USER-Access client (Initiator) across the network.

Once the USER-Access server process is started and communicating with the USER-Access client on a remote node, SI no longer plays a role in the connection. It does, however, keep the pseudo terminal (pty) resources locked up for the duration of the session so it can receive a "death of the child" indication when the session is complete. Upon receiving this notification, SI removes the child's entry from the /etc/utmp file (originally made by /bin/login).

Stopping the Service Initiator

The Service Initiator can be stopped by using the Service Initiator Control program /usr/nsc/sicom/si/control. Generally, the command to stop the Service Initiator is:

```
$ /usr/nsc/sicom/si/control STOP -H host  S service
```

where 'host' is the name of the local network host and 'service' is the name of a network service being offered by the Service Initiator ("USER" is generally the service name being offered). The optional OPERATOR password may need to be specified with the " O" option. Stopping the Service Initiator this way will clean up /etc/utmp entries without affecting the associated servers that are still running. Stopping the Service Initiator with 'kill' will cause those /etc/utmp entries to remain in the file until reboot or until they are reused.

The log file may be examined by issuing the following, while the Service Initiator is active:

```
$ tail /usr/nsc/sicom/si/svcinit.log
```

Providing a guest login

The Service Initiator provides for a default (or guest) login when no username and password are provided on the USER-Access CONNECT command. If you want a guest login for USER-Access, edit the file /usr/nsc/sicom/si/startup. Remove the comments (#) from the following lines:

```
#     USERNAME    guest
#     PASSWORD    secret
```

Change the username 'guest' and the password 'secret' to correspond to a valid username/password for your system. In doing so, this username/password account becomes a USER-Access guest account. Alternatively, add 'guest' to your list of valid accounts. The username and password you specify becomes the default if none are given on the CONNECT command.

The Service Initiator must be stopped and restarted for this change to take effect.

Service Initiator Keywords

The Service Initiator startup input file /usr/nsc/sicom/si/startup contains a list of keywords that can be set to alter the way the Service Initiator operates for a given SERVICE being offered.

SERVER Specifies the command that is used to invoke or run the USER-Access server.

OPERATOR Specifies a password that is required when issuing commands through the CONTROL program (used for shutting down the Service Initiator).

LOGTIMEOUT Specifies the login timeout in seconds. This is used to terminate a login request that for some unknown reason, is hanging around.

MINIMUM Specifies the minimum session number that will be offered for this service. For example, a MINIMUM value of 5 would result in SERVICE "USER" being offered as "USER005" up to MAXIMUM (below). Specifying MINIMUM or MAXIMUM does nothing to limit the number of USER-Access sessions that are allowed. They are simply provided as a naming tool.

MAXIMUM Specifies the maximum session number that will be offered for this service. A value of 30 for example would result in the last offer of "USER" being "USER030", before the offers started over at MINIMUM.

TRACE Allows different levels of tracing of the Service Initiator. Refer to the startup file /usr/nsc/sicom/si/startup for a description of the different trace levels.

UNIX_LOGIN This keyword and the two that follow (UNIX_PASSWD and UNIX_BADLOG) are to be modified only if the site has made modifications to the default UNIX login utility /bin/login. These three keywords define strings that the Service Initiator looks for when interacting with /bin/login during a USER-Access login request. UNIX_LOGIN defines the first prompt used by /bin/login when it is looking for a user name. The default value is the string "login: "

UNIX_PASSWD (see UNIX_LOGIN above). Defines the prompt used by /bin/login when it is looking for a password. The default value is the string "Password: ".

UNIX_BADLOG (see UNIX_LOGIN above). Defines the resulting string returned by /bin/login when an invalid login attempt has been made. The default value is the string "Login incorrect".

GATEWAY Enables/disables gateway processing of remote NODE qualifier. This should be 'on' for gateway hosts, 'off' for all others. DEFAULT: 'off'

SIHELPER Service name of the remote 'sihelper' process or NULL to disable SIHELPER support. DEFAULT: 'sihelper' to enable SIHELPER support.

SIPLUS Enables/disables SI+ proxy connect support. If 'on', the standard connect request is forwarded to the remote NODE. DEFAULT: 'on'

RLOGIN Enables/disables 'rlogin' to remote NODE. If 'on', the Unix 'rlogin' mechanism is used to connect to the remote NODE. DEFAULT: 'off'

SERVER Command invoked to activate the native network server. DEFAULT (Unix): exec /usr/nsc/sicom/user/start_server

UGSERVER Command invoked to activate the USER Gate server. DEFAULT (Unix): exec /usr/nsc/sicom/user/start_server_ug

RSERVER	Command invoked to activate the remote USER Gate server following a successful RLOGIN. DEFAULT (Unix): exec /usr/nsc/sicom/user/server
SIP_TIMEOUT	Connect timeout to use for SI+ proxy connects. This should be short to prevent long delays if the requested service is not offered on the remote NODE. DEFAULT: 1 sec.
SIP_INTERVAL	Connect interval to use for SI+ proxy connects. See also SIP_TIMEOUT. DEFAULT: 1 sec.
LOGINTYPE	The method used to log in to the system on which the Service Initiator is running. Valid values are 'rlogin' and 'login'. If 'rlogin' is specified, the rlogin protocol is used to log in to LOGINHOST. The 'login' method will allocate a pty, fork a process, invoke login, and invoke SERVER. DEFAULT: 'rlogin'
LOGINHOST	Host name or IP address to log in to when using LOGINTYPE 'rlogin'. DEFAULT: 'localhost'
OFFER_TIMEOUT	(NETEX only) The length of time (in seconds) to offer before timing out. If the offer times out, it is automatically reissued. DEFAULT: 0 (offer indefinitely).

USER-Access Site Startup Files

Site startup files are supported for the USER-Access client and server modules. These files allow the installer to establish site parameters (command defaults, alias definitions, etc.) for both the client (Initiator) and server (Responder). Two sample site files are provided with this release (in directory /usr/nsc/sicom/user):

```
sclient.ua    client (Initiator) site startup
sserver.ua    server (Responder) site startup
```

Site specific changes can be made to these startup files. An alternative to directly editing the client site startup (sclient.ua) is to create a site specific startup file named 'site.ua' in the same directory. If this file exists, it is automatically INPUT by 'sclient.ua'. Future updates of USER-Access may replace the 'sclient.ua' file (after making a save copy) but will leave the 'site.ua' startup file intact.

For example, the following lines could be added to the client site startup file:

```
# Site specific changes for USER-Access Initiator
#
set connect blocksize 32000
set alias printline text
```

The first command changes the default connect blocksize to 32000, and the second command defines an alias called 'printline'.

Any USER-Access command may be included in the CLIENT startup files. For security reasons, the SERVER startup file restricts the use of the commands: CONNECT, DISCONNECT, SEND, RECEIVE, LOCAL and REMOTE. Refer to the USER-Access Users Guide for more startup file information.

Update Summary

Release 5.0.1 M13 includes the following fixes and/or enhancements:

- For AIX: The DISTRIBUTION now contains three copies of most files, separated into three subdirectories. The AIX_V32 subdirectory contains files required for pre-4.2 installations (e.g., 3.x and 4.1). The AIX_V42 subdirectory is for 4.2 and 4.3 installations. The AIX_V5x subdirectory is for all AIX 5L installations. The install script and product file are still at the top level, thus installation procedures have not changed. The install script will detect which OS is running and will adjust accordingly. The install script will first issue a message telling the installer which OS was detected.
- For AIX 5.2: Setting the Service Initiator keyword LOGINTYPE to a value of "login" would cause remote connection attempts to fail with the message:

  ```
  /dev/ttyp0: 3004-004 You must "exec" login from the lowest login shell
  ```

  under the previous release. This problem has been fixed.

Release 5.0.1 M12 includes the following fixes and/or enhancements:

- For AIX: The DISTRIBUTION now contains two copies of most files, separated into two subdirectories. The AIX_V32 subdirectory contains files required for pre-4.2 installations (e.g., 3.x, 4.1). The AIX_V42 subdirectory is for 4.2 and 4.3 installations. The install script and product file are still at the top level, thus installation procedures have not changed. The install script will detect which OS is running and will adjust accordingly. The install script will first issue a message telling the installer which OS was detected.
- For AIX 4.2: Large file support was added to traditional USER-Access file transfers.
- For AIX 4.2: Large file support was added to the UABACKUP/UARESTORE utilities.
- (Unix) Previously, a remote login to the Service Initiator would sometimes fail with the message "Interrupted system call (UNIX-4)". This problem has been corrected.

Release 5.0.0 M11 includes the following fixes and/or enhancements:

- The following new CAM features are supported in this release. See the CAM Setup and Administration Guide for more information on each of these items:
  - Deleted File Support allows the ability to record names of files that have been deleted since the last backup has been added to CAM. This enables users to be able to restore a volume to the same condition it was at the time of a specified backup. Therefore deleted files are no longer restored unless requested. This feature requires the CAM Central Server to be at version 5.0.0 or later.
  - Informix Database backup (via the Informix onbar utility) provides full, incremental and transaction log backups of Informix databases (both scheduled and unscheduled) as well as client initiated restores. A new Unix utility type 'onbar' provides this service on selected Unix hosts.
- The following CAM Client bugs were corrected:
  - The Resilient Network Transfer (RNT) feature introduced in CAM 4.0 continuously tried to reconnect when a protocol error was detected. This problem has been corrected.
  - The Resilient Network Transfer (RNT) feature introduced in CAM 4.0 failed to properly support a secondary connection to the MVS Multiplexed Server (MUX). This problem has been corrected.

  - The *DIRECTORY line in the UABACKUP index file could be written twice when connected to a CAM 3.x Central Server. Duplicate top-level directories would appear when performing a selective restore. This problem has been corrected.
  - When archiving directories with ACLs assigned using the 'Select File List...' option, CAM would fail when attempting to capture the directory ACLs with the error 'Cannot retrieve ACLs for SSS (UAxxx-9437)'. This problem has been corrected.
  - The CAM GUI Client would sometimes crash when attempting to display an error message dialog. This problem has been corrected.
- (Motif) The font specifications in the CAM Motif Specification file were changed from 14-100 to 14-* allowing more host specific fonts to match.
- The Service Initiator now uses rlogin as the default login protocol.

Release 4.0.3 M9 includes the following fixes and/or enhancements:

- (Selected Unix hosts) Oracle Database backup (via Datatools SQL-Backtrack) provides full, incremental and transaction log backups of Oracle databases as well as client initiated object-level restores.
- (Unix) Previously, CAM Archiving would archive symbolic links by archiving the link information rather than the file (or directory) referenced by the symbolic link. Now, CAM archives symbolic links by "following the link" and archiving the file (or directory) referenced by the link.
- (Motif) Added support for displaying Japanese filenames. The CAM Motif specifications file has been updated to support Japanese (and other languages). The default font definitions in the 'defaults.cam' file will display Japanese fonts.
- A new 'locale' control file is used to map host dependent locale names to a CAM defined code set. A copy of this file exists in both the 'cam' and the 'user' subdirectories.
- A new *LOCALE keyword has been added to the CAM index files and the selection lists generated by the CAM Name Server contains a string value defining locale information (language, font, etc.) for the client host.
- Added support for a new filename escape character used to escape special characters (carriage return, line feed) in the CAM index file and Name Server selection lists. The new escape character is the Exclamation Mark (!) which does not conflict with any of the multi-byte character sequences used by Japanese, Chinese, etc. These changes should be backward and forward compatible.
- A new UARESTORE option -LIST will generate an index of files in the CAM container but will not restore the files. This option can be used to validate CAM container file data without performing an actual restore.
- (Unix) In the SCLIENT startup file, map the alias definition for BACKUP to UABACKUP instead of the previous 'tar' backup. The RESTORE, LIST and COPY aliases are also remapped.
- The following CAM Client bugs were corrected:
  - When processing a CAM GUI error dialog, the CAM GUI client would sometimes fail (core dump) due to uninitialized stack structures. This problem has been fixed.
  - The INPUT -SEARCH value was being forced to lowercase (for Unix). This was affecting pathnames with uppercase components. The pathname case is now preserved unless (SITE) or (USER) is specified.

- The *DIRECTORY line in the UABACKUP index file could be written twice when connected to a CAM 3.x Central Server. Duplicate top-level directories would appear when performing a selective restore. This problem has been corrected.
- When archiving directories with ACLs assigned using the 'Select File List...' option, CAM would fail when attempting to capture the directory ACLs with the error 'Cannot retrieve ACLs for SSS (UAxxx-9437)'. This problem has been corrected.
- The CAM GUI Client would sometimes crash when attempting to display an error message dialog. This problem has been corrected.

- Limitations of the USER-Access Remote environment may result in the error 'Overflow of 1000 byte environment buffer (UA-302)'. The Remote environment was increased from 1000 bytes to 2000 bytes to prevent this condition.
- (Motif) The CAM Motif client has been enhanced to provide a more 'user-friendly' interface when running in the Sun OpenLook environment. Previously, a stack of CAM dialogs could get sequenced out-of-order making it difficult to find the current 'active' dialog. Now, a stack of CAM dialogs is always kept properly sequenced with the 'active' dialog on top.

  Previously, a CAM child dialog always centered itself on the parent dialog. Therefore, if a CAM dialog was moved off of the visible screen, it's child dialog may be partially hidden as well. Now, child dialogs are always forced to display completely on the visible screen.

Release 4.0.1 M8 includes the following fixes and/or enhancements:

- In the previous release, compression of large files (greater than 1 Gbyte) could cause a divide-by-zero 'core dump' of the USER-Access Responder while performing a UABACKUP (or USER-Access file transfer). This occurred when resetting integer byte counters to avoid a numeric overflow condition. All byte counters and compression ratio calculations are now performed using double precision to avoid any possible numeric overflow.
- In the previous release, when UARESTORE (or USER-Access file transfer) was expanding previously compressed data, the expanded results could have been truncated. The original data usually contains large sequences of repeated characters (such as 'sparse' database files with large blocks of zeroes). The expand logic was corrected to completely flush the data stream to avoid truncation.
- (Motif) During initialization of a CAM dialog, the Central Server has the option of positioning the cursor to a field other than the first logical field. This feature could be used, for example, to skip over fields already containing data. Under Motif, this feature did not work correctly. This problem has been corrected.
- The following new CAM features are supported in this release. See the CAM User Guide for Unix for more information on each of these items:
  - CAM Archiving allows selected sets of files/directories to be archived using a list of specifications stored in a file. A CAM Name Server is supported on this client. The Name Server is required for point-and-click selection of archive files.
  - CAM Resilient Network Transfer (RNT) automatically recovers from network failures from the point of interruption. The network connection is reestablished following a dropped connection, a checksum or CRC failure as well as most other network failures.
  - (Selected Unix hosts) Sybase Database backup (via Datatools SQL-Backtrack) provides full, incremental and transaction log backups of Sybase databases as well as client initiated object-level restores.

- Resizing of the CAM client main window is now supported. Button positioning is determined dynamically.
- Following a successful CAM connect, the Help->About... dialog displays the Central Server Hostname.
- When storing filenames in the UABACKUP index file, embedded carriage return and linefeed are converted to the character sequence `CR and `LF. Embedded escape characters (backward quote) are double escaped ``.
- A new CART tape label qualifier STREAM tells CART to process a stream of data with no tape label processing. This allows CART to operate on data where the operating system performs the tape label processing. CART can also read disk files or piped data using the STREAM tape label option.
- The USER-Access INPUT command supports a new Boolean qualifier -STRIP. A -STRIP value of 'on' strips trailing blanks from USER-Access input lines.
- Changed the default Maximum Line Count in the File->Preferences dialog from 1000 to 500 to prevent CAM list box overflows and improve list box performance. Changed the default Printer Lines Per Page from 54 to 60 to better fill a standard print page.
- All CAM byte counters are now handled as double precision numbers to support transfers larger than 4 Gbytes. All byte count displays will convert to kilobytes (K), megabytes (M) and gigabytes (G) as needed.
- The CAM client help file 'clihelp.cam' has been updated.
- Internal buffer sizes were increased in the USER-Access Initiator and the CAM Responder to support long filenames. The length of USER-Access aliases has been doubled to 2000 bytes.
- The minimum USER-Access connect blocksize was changed from 1024 bytes to 2048 bytes to handle the larger internal record sizes.
- (Unix) The CAM 'install' script checks for a pre-existing /.uabackup file. The user is allowed to keep the existing file or create a new privileged user list.
- (Unix) A new CAM icon for the Motif GUI interface has been added. This icon appears when the Motif GUI client is minimized.
- The following bugs have been fixed in the CAM Client code:
  - Previously UABACKUP would log a *ERR record in the index file following a file open failure. Now the open is attempted before any index entry is logged. If the open fails, the failure is reported in the exception log but no entry is made in the index or container file.
  - Previously *SEGMENT records were not written correctly to the UABACKUP index file if the file spanned a container file segment and was the last file in the directory. This problem has been corrected. Note: CAM Central Server logic was also enhanced to handle this condition.
  - Compression of files containing long sequences of repeated characters (such as 50 Mbytes or more of 'sparse' data containing all zero bytes) would fail to expand properly. This problem has been corrected.
  - A program abort could occur with 'debug protocol' enabled. This problem has been corrected.
  - (Unix) Directory create/modify times are now preserved for a CAM UARESTORE.

    - (Unix) Hard links to Unix device nodes are now restored properly. Previously a duplicate device node was created instead of a hard link.
    - (Unix) Zero length files following a Unix 'sparse' file were restored with an improper file length. This problem has been corrected.
    - (Unix) Closing and reopening directories during UABACKUP could cause directory positioning failures and possible looping. Directories are now left open when traversing subdirectories to avoid this failure.
- Fixed a bug causing a segmentation fault when the USER-Access client was started with an invalid input file on the command line.
- Added support for an OFFER_TIMEOUT keyword in the SI startup file. If the NetEx offer times out, it is automatically reissued. This keyword is processed for NetEx networks only.
- The CAM graphical client error dialog box now displays a variable title, “CAM sss Message”, where sss is either Fatal, Error, Warning, or Informational depending on the type of message displayed. The dialog is also wider, and supports a minimize button.
- The install procedure has been modified in many ways:
    - The installation must now be performed as root. The installation script will prompt for a Unix UserID to serve as the owner of all installed files.
    - A USER Gate installation now provides executables for both USER Gate and TCP/IP protocols. This is done to support the new Service Initiator proxy connections.
    - The “CAM” installation option is provided to install only the CAM Motif graphical client and associated files.
    - Privileged Responder configuration for CAM client hosts is now an installation option.
    - Executable scripts are no longer placed in the /usr/bin directory (or a selected alternate 'bin' directory). Instead, the scripts are kept with the associated executables, and symbolic links are added to a 'bin' directory selected at installation time (the default is still /usr/bin).
    - Previous installations required a manual edit of the Service Initiator startup file if an alternate 'bin' directory was chosen (see item 'e' above). This is no longer required.
- UABACKUP and UARESTORE have been modified to support changes in the CAM architecture:
    - A -STRIP option has been added to UARESTORE. During a partial restore, this qualifier controls how much of the parent directory structure is built for a restored file.
    - The include/exclude lists can now be given as "<filename" where filename is the full path name of a file containing a list of patterns. For CAM graphical client users, this syntax can be used in the Exclude field for a volume. The filename given must exist on the client host during the backup of that volume.
    - Members of an exclude list can now be full (absolute) pathnames.
    - For UABACKUP, directory information for directories that do not match the backup criteria will not be backed up. Previously, all directory information for a volume was always backed up.
    - For UARESTORE, error messages are now written to the Log file and the Error Log.
    - For UNIX: Added special handling of hard links and symbolic links.
- For UNIX: SEND/RECEIVE qualifiers -DIR_CREATE and -DIR_OPEN have been added. These are for use by the CAM Server.

- A problem would occur when the INPUT qualifiers -IGNORE and -SEARCH were used together. This bug has been fixed.
- A "uarestore" alias has been added to the sclient.ua file to support CART restores of CAM backups. See "Using the CART Utility with CAM" in section VI, "Documentation Updates," for more information.
- The CAM Motif graphical client will force a reload of the default resource file if the transfer of a Central Server resource file fails. Previously, the GUI client would get trapped in a loop, trying to display an error message, but unable to do so due to a lack of resource information.
- In the previous release, a Privileged USER-Access Server could potentially fail to validate a CAM backup user ID. The Server would err conservatively and would revoke privileges for a valid backup user ID (i.e., the problem with the previous release did not result in a security gap). This problem has been fixed.
- UABACKUP no longer attempts to back up active Unix sockets. Because a Unix socket is associated with a running process, there is no reasonable way to restore the node. It is therefore a mistake to back it up.
- Previously a subset of the Administrative HELP text was provided for CAM Users. However this eliminated some necessary User HELP text. Now all HELP text is available to Users.
- Corrections and additions were made to the CAM client and CAM server HELP files. Conditional text is included for MVS tape/disk classes and Unix Unitree.
- CAM client: Underscores typed in an edit field did not display properly. The XmText marginHeight was set to 0 to allow underscores to display.
- CAM client: Updating of list boxes performs very poorly (especially for Select All and Clear All requests). This has been improved somewhat.
- Changed the CAM HELP file format allowing help text to continue on multiple lines with each line less than 72 characters. A number of CAM HELP bugs were also corrected. If trailing blanks were stripped from the help text files, the GUI display would run words together. The NEXT button did not dim when the end-of-list was encountered. New CLIHELP.CAM and SERHELP.CAM files were provided.
- The DEFAULTS.CAM option PRINT_AUTO_FF was added to force a trailing form feed at the end of printer output.
- Fixed a bug in the Privileged Server code which failed to allocate the global privileged structure.
- Increased the size of the login prompt buffer from 256 characters to 1024 characters to prevent an overflow when processing login output. Long output lines could cause the Service Initiator to crash.

Release 3.0.6 M7 includes the following fixes and/or enhancements:

- In the previous release, UARESTORE would not restore the high-order mode bits (setuid, setgid, etc.). This problem has been fixed. Since the bug was in UARESTORE and not UABACKUP, CAM backups taken with the prior UABACKUP utility can be restored correctly.
- In the previous release, if UARESTORE were to restore a sparse file followed by one or more zero-length files, those files would acquire the length of the sparse file. This would continue until a file with a non-zero length was encountered by UARESTORE. This problem has been fixed. Since the bug was in UARESTORE and not UABACKUP, CAM backups taken with the prior UABACKUP utility can be restored correctly.

Release 3.0.6 M6 includes the following fixes and/or enhancements:

- The UABACKUP/UARESTORE utilities have been modified to back up and restore the Access Control List ("ACL") information on files containing extended ACLs.
- In the previous release, if UABACKUP ran without privilege and encountered a directory that it did not have permission to access, the log file (a.k.a., "index file") would contain erroneous information. For the CAM product, this would result in erroneous displays of file locations when a user attempted to select files for restore. This problem has been fixed.
- The CAM graphical client's Motif defaults file, "CAM," has been modified so that underscores display correctly.

Release 3.0.6 M5 includes the following fixes and/or enhancements:

- RECORD-mode file transfers can now handle a piped source specifier. For example, the following is legal from the USER-Access Client: SEND -MODE RECORD "!record_generator" recfile
- The previous version could not properly handle a file transfer using both file segmentation and CRC. This problem has been fixed.

Release 3.0.3 M4 includes the following fixes and/or enhancements:

- The input line size has been increased from 500 characters to 1000 characters. This allows for larger alias definitions.
- The token size has been increased from 200 characters to 500 characters. The token size affects the size of quoted strings such as those used for piped commands.
- An "Accumulated:" record was added to every DEBUG INTERVAL display. If the DEBUG INTERVAL is set during a file transfer, this new record will be displayed at each interval. It contains the accumulated transfer size and the average transfer rate.
- The Service Initiator on a USER-Gate gateway can now be configured to log in to a remote TCP/IP node via the Service Initiator on that node, as opposed to using the "rlogin" or SIHELPER methods. See the section entitled "SERVICE INITIATOR KEYWORDS" in this Memo for more information.
- Unix: The Service Initiator has been modified to periodically discard any pending data on the pseudo-terminal allocated for each active SERVER. This prevents the SERVER from hanging upon termination if some output has been directed to that pseudo-terminal. Such output could come from "write" or "wall" commands. Also, MLS security login processes tend to output session statistics when the user logs out (i.e., when the SERVER terminates).
- Unix: USER-Access is now linked with the host's linker, not with the compiler. A "C" language compiler is no longer an installation requirement. Also, as part of the installation, Bourne shell scripts needed to invoke USER-Access executables are placed under /usr/bin (or another site-selected location).
- The string displayed by the command "SHOW LOCAL VERSION" has changed to match the product's new version numbering system. Note that the "R-number" has been dropped. Delivered USER-Access scripts that checked this string for a particular range of R-numbers have been modified. Any site-specific USER-Access scripts that manipulated the version string may need to be altered.
- The CAM Client has been added to the standard product. This executable provides a Graphical User Interface for remote administration of the Central Archiving Management (CAM) system (a separate product).

- Unix: Added Privileged-Server capabilities. If the SERVER (or CLIENT, for that matter) has SETUID privilege, and the user's login name appears in the file "/.uabackup", then the SERVER will run with root privs. This allows privileged file transfers, including UABACKUP functions. See section VI, "Documentation Updates," for more detail.
- The server command line option PASSWORD can now be a locally encrypted password.
- Support for segmented file transfers has been added. This feature is provided for CAM backups, and is not recommended for general use.
- Internally processed piped commands UABACKUP and URESTORE have been added.
- Added Smart-Restore capabilities. The sending side can filter a UABACKUP container file and thus send only the portions required. Filters include lists generated by CAM, or filters given in the returned environment from URESTORE initialization.
- USER-Access script functions XFER("DATE") and XFER("SEGMENT") were added.
- Unix: If USER-Access is installed for a NETEX network, the USER-Access installation script will now attempt to locate the NETEX interface library to be used. The script will first try to locate /usr/lib/libntx.a, and if that library is not found, the script will use /usr/lib/libntxuser.a.

Release 2.0 R10.3 M3 includes the following fixes and/or enhancements:

- The previous release had a problem when using the -EXPand qualfier on two simultaneous transfers. When the first transfer terminated before completion, either with a keyboard interrupt or network failure, the second transfer using the -EXPand qualifier would fail.
- A checksum problem occurring on machines with non-VAX byte ordering was fixed.
- The previous release had a problem when a second TCP connection failed. The first connection would be disconnected as well. This problem has been fixed.
- The previous release for RISC/6000 AIX had various related problems during login through the Service Initiator: (a) The Service Initiator would refuse logins if its controlling terminal was logged out; (b) remote logins (i.e., using the -NODE qualifier with the CONNECT command) would not reliably detect prompts issued by the remote system; (c) a remote login could potentially lock up the Service Initiator, preventing further logins. These problems have been fixed.

Release 2.0 R10.1 M2 includes the following fixes and/or enhancements:

- Data compression support has been added as a SEND/RECEIVE option for file transfer and central archiving. The qualifiers COMPRESS, EXPAND and METHOD have been added. See section VI (Documentation Updates) for a complete description of this new feature.
- A new checksum option has been added to increase data transfer performance while maintaining data integrity. The checksum algorithm is the 16 bit Internet checksum (used by IP, UDP and TCP). In most host environments the checksum algorithm performs much better than the 32 bit CRC. A new SEND/RECEIVE Boolean qualifier CHECKSUM enables this option. If both CRC and CHECKSUM are enabled, the CRC option overrides the CHECKSUM option producing a full 32 bit CRC calculation.
- Performance improvements were made to the TCP/IP version of USER-Access by reducing the number of system I/O requests during file transfer. Previously the block header and associated data were handled with separate read/write requests. A single read/write request is now issued for each network block.

- Support for tab expansion has been added to CHARACTER mode transfers. A new qualifier TAB specifies the numeric tab stop (eg. TAB 8). A value of zero (the default) disables tab expansion. Tab expansion must be supported by the RECEIVING host machine. Some hosts (especially Unix hosts) pay a significant performance penalty to expand tabs. However TAB tends to be faster than a similar approach using command piping. In addition, wildcarding is supported with TAB and not with command piping. Tab expansion is not supported for V1CHAR mode.
- The default prompt for the standalone USER-Access SERVER was changed from 'Server:' to '{time()}:' causing all standalone SERVER output to be time stamped.
- A new local/remote GATEWAY qualifier was added. Following a successful USER-Gate connection, GATEWAY is defined as the official name of the gateway host. The gateway for the USER-Access client appears as a LOCAL qualifier. The gateway for the USER-Access server appears as a REMOTE qualifier. Non USER-Gate clients/servers will show GATEWAY as NULL (undefined). This can be tested using USER-Access string functions for USER-Gate sensitive scripts.
- File transfer mode RESTORE has been enhanced to allow hosts with different byte ordering to properly decode the length field defined by the archive file format. Previous versions would fail with the error: Invalid ARCHIVE block length (nnnn) (UA 5305). This addition allows hosts with different byte ordering to share archive files (most likely Unix hosts).
- Added a DELETE_ON_ERROR boolean qualifier when transferring a file. If DELETE_ON_ERROR is set to 'on', the destination file is deleted if an error occurred during the file transfer. This eliminates partial or incomplete files following a failed transfer. The default DELETE_ON_ERROR setting is 'off' for compatibility with previous releases.
- Previous releases of USER-Access forced an error when referencing an invalid command qualifier during string substitution. For example, {bad:local} would generate the error: Invalid LOCAL qualifier 'bad' (UA 5408). This string substitution error has been eliminated. Any invalid command qualifier is treated as undefined, substituting a null string. This allows backward compatible USER-Access scripts to test for the existence of a new command qualifier. For example, the following will work properly without generating an error.

  ```
  text {dfn(bad:local, "defined", "not defined")}
  ```

- When connected (via Netex) to a non-8-bit host (eg. CDC/NOS, OS 1100) any unused bits in the last byte of a binary block are now cleared to zero. Previous releases left possible garbage bits in the last byte of a binary block.
- CAP0 and CAP1 are now valid LOCAL qualifiers. They can be displayed (eg. SHOW LOCAL CAP1) or referenced as strings (eg. {cap1:local}) for testing specific capabilities. The following example tests for COMPRESSION support (CAP1 capability H1):

  ```
  text {ne(index(cap1:loc, "H1"), 0, "COMPRESSION is supported")}
  ```

- Square brackets in USER-Access error messages have been replaced by single quotes. This eliminates some translation problems when messages are sent to EBCDIC machines where square brackets are typically not supported. This also avoids some translation problems in Europe where national characters replace square brackets.
- Added a new INPUT qualifier -IGNORE used to ignore the failure condition if the input file does not exists. The default setting is 'off' causing INPUT failures to be reported. This qualifier is used in the 'sclient.ua' startup file to input a site specific startup file using the command:

  ```
  input -ignore {rootdir:local}site.ua
  ```

- If the 'site.ua' file exists, it is processed. If the file does not exits, the error is ignored. This allows a site to isolate their startup file changes into the file 'site.ua'. The 'site.ua' file will not be altered as part of a USER-Access update.
- For UNIX: USER-Access LOCAL/REMOTE command execution no longer invokes .profile as part of Bourne shell startup. The .profile startup file is usually processed at Service Initiator login time.
- For UNIX: USER-Access piped command execution did not handle terminal prompts when piping to STDIN running under the Bourne shell. These prompts were directed to the /dev/tty device. Unlike the C-Shell, the Bourne shell does not change the process group when executing a command. USER-Access now forces a process group change during piped command execution.
- For UNIX: If a USER-Access client is started with SIGINT (keyboard interrupts) ignored, they will continue to be ignored by USER-Access. This allows background jobs running under the Bourne shell to ignore keyboard interrupts. The C-Shell uses a different mechanism for background jobs so keyboard interrupts were already ignored.

Release 2.0 8905 R9 M1 includes the following fixes and/or enhancements:

- The Service Initiator along with its startup procedure and control program are part of this release. Refer to Step 5 of the installation instructions and the section entitled "AUTOMATICALLY STARTING THE SERVICE INITIATOR".

Release 2.0 8905 R9 M0 is the initial release of this product.

Documentation Updates

Running a privileged USER-Access responder

For CAM client installations that do not want to perform backups under the root user account, the USER-Access install script will optionally prompt for the name of a designated backup UserID. The Unix UserID selected will be able to run the USER-Access Responder with the privileges necessary to back up all of the system's volumes via the CAM uabackup or uaraw utilities. (Note that this feature does not apply to sites running traditional USER-Access CAMTools on satellite hosts.) A recommended UserID would be one created specifically and solely for CAM backup and restore activities.

If requested, the install script will create a file named "/.uabackup" containing the name of the selected backup UserID, and will set the file's permissions to 0600. The ownership of the USER-Access Responder is then changed to root, and the SETUID permission bit is set for this executable. This allows the Responder, which is started by the Service Initiator upon a request for service USER, to start up with root privileges and to read the "/.uabackup" text file. If the user name used to log in via the Service Initiator is listed in the file, the Responder will continue to run with root privileges, otherwise the extra privileges will be dropped immediately, and the Responder will run with the user's login privileges.

The Privileged Responder allows the UNIX user listed in the "/.uabackup" text file to perform file transfers with root privilege, including the use of the USER-Access internally processed piped commands, uabackup and uarestore. Remote command execution and external-pipe file transfers (e.g., send tarfile "!tar xvf") are performed with the user's login privileges, not with extra root privileges.

If this option is not chosen at the time of the installation, the steps necessary to set up a privileged Responder can be performed manually at a later time:

1. The USER-Access server executable must be owned by root, and must have the SETUID permission bit activated. To do this, you must log in as root and enter the following commands (if yours is a USER Gate network installation, use "server_ug" in place of "server"):

```
chown root /usr/nsc/sicom/user/server
chmod u+s /usr/nsc/sicom/user/server
```

2. Create a file called ".uabackup" directly under the root directory. This text file must contain the privileged account name on a single line, without extraneous white-space.

3. Make sure the /.uabackup file has root ownership, and root is the only account capable of reading or writing this file:

```
chown root /.uabackup
chmod 600 /.uabackup
```

Although the install script will only write one UserID to the /.uabackup text file, you may add as many privileged names as necessary, one per line (the name "root" should not be included). Account names can be added and removed at any time using any text editor, but the file can only be edited by root.

The CAM client defaults files

Note: This section applies only to CAM Client sites. It does not apply to sites running traditional USER-Access CAMTools on satellite hosts.

If a user of the CAM graphical client executable specifies some Preferences, a file named defaults.cam will be created in the user's home directory. This text file contains CAM Preference variables and their values, as defined in "USER-Access CAM Central Archive Management User Guide for UNIX, Appendix C".

Default parameters for the CAM Client Graphical User Interface display, such as font and color selections, are stored in the file ./cam/CAM under the TARGET directory.

Password encryption

The string function ENCRYPT has been added to USER-Access. The purpose of this function is to encrypt host passwords which later will be used by USER-Access to establish host connections. This approach eliminates the security risk of having readable (clear-text) passwords stored in files.

Format:

```
encrypt(password, [username])
```

Where:

password — Specifies the password you want to encrypt. The encrypted form of this password is returned by the ENCRYPT string function. The encrypted form can be stored in script files containing USER-Access CONNECT commands.

username — Optionally specifies the username associated with the local USER-Access process that will issue the CONNECT command. This username is used as a secondary encryption key for the specified password. When USER-Access is later run it queries the operating system for the username running the current process. USER-Access then uses this username as one of its keys in decrypting the password. A value of '*' (single asterisk) tells the USER-Access ENCRYPT function to use the current username running the USER-Access process as the secondary key. Note that the effective username is used on UNIX. You must be running as the same user which will later run USER-Access to issue the CONNECT command.

For example, encrypt the password 'COBRA' using the UNIX username 'myers' as the local username for secondary encryption. Use the USER-Access TEXT command to display the encrypted results:

```
User> text {encrypt("COBRA", "myers")}
User: *249eece8e4203b189
```

The ENCRYPT ALIAS

To simplify the use of the ENCRYPT string function, an ENCRYPT alias is provided in the SCLIENT startup file in the USER-Access distribution. The ENCRYPT alias definition is shown:

```
set alias ENCrypt {} {dfn(1, "goto skip")} !
      ask -secure -prompt "Enter password? " 1 !
      ask -prompt "Enter optional username (or '*')? " 2 !
      skip: set global pw {encrypt(1, 2)} !
      text The encrypted password is {pw}
```

For example, the ENCRYPT alias could be used to encrypt the same password 'COBRA' with the same secondary key 'myers' shown previously:

```
User> encrypt
Enter password? COBRA   (password does not display)
Enter optional username (or '*')? myers
User: The encrypted password is *249eece8e4203b189
```

Note the following items regarding the ENCRYPT alias:

1. The password is prompted in -SECURE mode to avoid displaying on the terminal.
2. The ENCRYPT alias can be invoke with 'password' and optional 'username' passed as alias parameters to avoid prompting. However, the password will display.

3. The optional 'username' is forced to uppercase using the UPPER string function.

4. The resulting encrypted password is stored in a global variable PW for later reference.

Examples

Example 1: Encrypting Passwords Stored in a USER-Access Input Script File

Suppose a job running under the local UNIX username ‘nscjones’ inputs the USER-Access script ‘mvs1.ua’ during program execution, and the script ‘mvs1.ua’ contains the following line:

```
CONNECT mvs1 admin7 secret
```

To avoid storing the password ‘secret’ in readable form in the script file, the password is encrypted by invoking the USER-Access client and using the ENCRYPT alias:

```
User> encrypt secret nscjones
User: The encrypted password is *26f17e2a4c9c65c56
```

Username ‘nscjones’ is specified because that is the local UNIX username under which the USER-Access job that uses the connect/login information will run. Using a local text editor, modify the input script ‘mvs1.ua’ to look like:

```
CONNECT mvs1 admin7 *26f17e2a4c9c65c56
```

Example 2: Using USER-Access to Generate the Input Script File

As you can see in the ENCRYPT alias definition, the global variable ‘pw’ is set to the encrypted password value. This value can be used to generate an input file containing the USER-Access CONNECT command to be later referenced by a USER-Access script. We can use the USER-Access OUTPUT command to generate the script file ‘mvs1.ua’ to connect to the host ‘mvs1’ as user ‘admin7’ with the password ‘secret’ (as shown in example #1):

```
User> encrypt secret nscjones
User: The encrypted password is *26f17e2a4c9c65c56
User> set output prefix
User> output mvs1.ua
User> text CONNECT mvs1 admin7 {pw}
User> output
```

The resulting file mvs1.ua will contain the following line:

```
CONNECT mvs1 admin7 *26f17e2a4c9c65c56
```

Example 3: Encrypting Passwords Stored in a Central Archive Management Tools Configuration File

The Central Archive Management Tools require that each satellite site create an 'hconfig' file that contains the local username and password by which the central site backup job can access files on the satellite site. All satellite site 'hconfig' files are stored at the central site. The following is a sample 'hconfig' file:

```
* This is a sample HCONFIG file for satellite host TAN01
*
* The following is the network host name
      sat_host    tan01
*
* The following is the username/password to use for backups.
* This account has suitable privileges to read volumes.
      sat_username      mr_backup
      sat_password      safety
                  .
                  .
                  .
```

To avoid storing the password in readable form in the 'hconfig' file, the password can be encrypted using the USER-Access client as shown:

```
User> encrypt safety MVSADMIN
User: The encrypted password is *27e0117547d7a8486
```

In this example, username MVSADMIN is specified because that is the central site username under which the backup jobs are run. Using a text editor, the user enters the encrypted form of the password in the 'hconfig' file:

```
* This is a sample HCONFIG file for satellite host TAN01
*
* The following is the network host name
      sat_host    tan01
*
* The following is the username/password to use for backups.
* This account has suitable privileges to read volumes.
      sat_username      mr_backup
      sat_password      *27e0117547d7a8486
```

Keep in mind that the 'hconfig' file is used on the central site host. Therefore any changes made on the satellite system would require that this file be re-sent to the central site. Most likely, the central site administrator would run ENCRYPT centrally and edit the 'hconfig' file there rather than on the satellite. However, the procedure for getting this accomplished is as described in the example.

USER-Access Data Compression

Support has been added to USER-Access for data compression and expansion during file transfer. The new SEND/RECEIVE qualifiers are:

COMPRESS	- compress the source data stream (on/off)
EXPAND	- expand the destination data stream (on/off)
METHOD	- the method of compression (LZW, RLE)

The two compression methods currently supported are LZW and RLE. The LZW method uses the Lempel-Ziv-Welch algorithm for finding common substrings. This method is deterministic and can be performed on the fly. Block compression is performed with an adaptive reset whereby the code table is cleared when the compression ratio decreases. This method generally provides the best overall compression ratio, but requires significantly more CPU resource than the RLE method. LZW compression ratios for character data are typically 40% to 60% of the original data size. LZW compression ratios for binary data are difficult to predict. Applying LZW compression to already compressed data could actually increase the data size up to 130% of the original size.

The RLE method uses a simple Run Length Encoding algorithm that counts strings of repeated characters (usually spaces or nulls). This method is much faster than LZW compression but provides much poorer compression ratios (typically 80% to 95%). Unlike LZW, the RLE method will never grow data that is already compressed (except for the addition of the compression header).

The following examples demonstrate file transfers using the -COMPRESS and -EXPAND qualifiers.

Send a binary source file ‘data’ with data compression enabled. The destination file ‘data.cmp’ contains the compressed data:

```
User> send -mode stream data data.cmp -compress
```

Receive the same compressed file expanding the data stream back to the original binary file:

```
User> receive -mode stream data.cmp data -expand
```

The same binary data file can be compressed, sent across the network and expanded into the destination file:

```
User> send -mode stream data -compress data -expand
```

One-sided compress/expand (the first two examples) is possible when connected to earlier releases (pre-R10) of USER-Access for all supported modes except CHARACTER. Two-sided compress/expand (the last example) requires that both sides (client and server) support compression.

Only certain combinations of -COMPRESS and -EXPAND are valid with the various USER-Access transfer modes. The following table shows which combinations are valid (Yes) and which are not valid (No):

Transfer Mode	-COMPRESS only	-EXPAND only	Both -COMP/-EXP
CHARACTER	Yes	Yes	Yes
RECORD	No	No	No
STREAM	Yes	Yes	Yes
BACKUP	Yes	No	Yes
RESTORE	No	Yes	Yes
COPY	No	No	Yes
V1CHAR	No	No	No

Character mode compression

Both sides (client and server) of a CHARACTER mode transfer must support compression (release R10 or later). In addition, when transferring between hosts with different native character sets (e.g., ASCII to EBCDIC) there are some subtle problems caused by the fact that only the USER-Access client performs code conversion.

The character set of the compressed data is stored in the compression header that prefixes the compressed data stream. This information can be used during expansion to determine if code conversion must be performed.

The following table illustrates the various combinations of CHARACTER mode compress/expand. The source and destination file types are shown as well as any code conversion issues:

USER-Access command	Source	Destination	Code Conversion performed
send-compress	text	stream	by client before compress
receive -compress	text	stream	not done - server pushes an informative message - flags its native char set in header
send -expand	stream	text	not done - error if char set in header does not match server's char set
receive -expand	stream	text	by client after expand if char set in header does not match client's native char set
send -compress -expand	text	text	by client before compress
receive -compress -expand	text	text	by client after expand

Central Archive Restore from Tape (cart)

Overview

The CART utility allows a satellite system user to restore previously archived files (via Central Archiving) from a central host in the event of a disabled network or unavailable central host. CART makes it possible to restore UNIX files that reside on a central host's disk or tape, without requiring a functioning network or central host. With CART, Disaster Recovery is possible since an entire UNIX system can be restored at some off site location, independent of the network. Although any USER-Access host can act as a central site for archiving files, an IBM MVS system is used here for example purposes. The concepts discussed below regarding the central site can be applied to most any system (e.g. IBM VM, Unisys OS1100, VAX/VMS, UNIX, etc.).

Assuming a UNIX system has been backed up via Central Archiving to an IBM MVS central site, the general concept behind restoring the UNIX files in the event of a disabled network is as follows.

(1) Copy the previously archived UNIX container file from IBM DASD or cartridge tape to 9-track tapes using standard IBM utilities.

(2) Move the 9-track tapes to an existing UNIX tape drive.

(3) Invoke the CART utility to read the tapes and stream the data into the UNIX restore utility (tar, cpio, restore, etc.), restoring the selected files. CART handles various types of tape labels (e.g., IBM, ANSI, NONE). CART is needed to read through the Central Archiving header and trailer information which is part of all container files.

The sections that follow describe the process of copying UNIX container files (residing on an MVS host) to 9-track tape and restoring them using the CART utility. It is assumed the reader is familiar with USER-Access and Central Archiving.

Copying a Centrally Archived Container File to 9-Track Tape

Note: The procedure described below is appropriate for restoring container files generated with USER-Access Central Archiving or with the Central Archiving Management Tools. A similar procedure for the container files generated with the CAM UABACKUP is described in "Using the CART Utility with CAM" later in this document.

Most Central Archive sites backup their UNIX files directly to IBM MVS 3480 cartridge tape. Others will backup directly to DASD. Whichever is the case, in order to restore the backed up data to the UNIX system in the event of a down network, the archived container files must first be copied to a medium that can also be read by the UNIX system. Since 9-track tape is the most common, the focus of attention here will be dedicated to it. (Sites backing up directly to IBM 9-track tape may use those tapes directly with CART).

The method for copying a previously archived UNIX container file to 9-track tape is basically the same whether it resides on cartridge tape or DASD. The following is sample IBM JCL that invokes the IEBGENER utility to copy a container file stored as data set CA.UNX1ROOT.JAN2290 on cartridge tape, to a new data set named UNX1ROOT.JAN2290 on 9-track tape:

```
//NUACOPY  JOB ,'UA TAPE COPY',CLASS=A,MSGCLASS=X,
//            NOTIFY=NUAUSER,COND=(0,NE)
//*
//*  REFER: NUA.V2R0M6.TEXT(NCACOPY)
//*  COMP:  IBM/MVS USER ACCESS (H213)
//*  DOC:   INVOKE IEBGENER UTILITY TO COPY A CENTRAL ARCHIVE TAPE
//*
//COPYTAPE EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=CA.UNX1ROOT.JAN2290,DISP=OLD
//SYSUT2   DD DSN=UNX1ROOT.JAN2290,DISP=(NEW,KEEP,DELETE),
//            DCB=(CA.UNX1ROOT.JAN2290,DEN=3),
//            UNIT=TAPE,LABEL=(1,SL)
//*
//LASTSTEP EXEC PGM=IEFBR14         THIS PUTS COND CODE IN NOTIFY
```

The data should be copied to 9-track tape at a density that can be read by the UNIX tape drive which will be performing the restore. Notice the DCB parameter DEN above. A DEN value of 3 represents 1600 bpi whereas a DEN value of 4 represents 6250 bpi. Also, it is recommended that the data set DCB characteristics (such as LRECL, RECFM, etc) be left intact as much as possible when performing the copy. Since the data within the data set is a UNIX backup utility save set (the output of tar, cpio, dump, etc.), it should not be modified during the copy or the utility will reject it during restore.

Generally procedures to copy data sets from one device to another are already in place at most sites. The IBM operations staff will most likely recognize this step as a standard utility procedure.

Note: If the container file has been segmented, the segments must be appended to form a single dataset on the tape volume set. The section titled Using the CART Utility with CAM later in this document contains example JCL showing how several segments can be appended into a single tape container file.

Running the CART Utility

The CART utility is distributed as part of the general USER-Access release. This section assumes the UNIX system being used to do the restore already has USER-Access installed. If this is not the case, see Installing CART on a non-USER-Access System before continuing.

Prior to running CART, the 9-track tape containing the UNIX container file (created in the previous section), should be physically mounted on a local UNIX tape drive. The tape drive should be set online. This example assumes USER-Access has been installed in the suggested directory /usr/nsc/sicom. To invoke CART, issue the following command:

```
$ /usr/nsc/sicom/user/cart
User>
```

By default, CART, a modified version of USER-Access, reads the USER-Access site and user startup files. The startup files allow users to pre-define aliases and command defaults that may be used later during CART processing. Within these files, it is typical to define the prompt string to “User>”, which is the case in the example above. The prompt string at any given site will vary based on how the startup files are configured. Regardless of the prompt, the basic operation of CART is the same.

There are two new commands that CART introduces, ATTACH and DETACH. CART also adds new qualifiers to the RECEIVE command which is used to perform the restore. The new qualifiers are REWIND and SKIP. The new commands along with the new RECEIVE qualifiers are described in “CART

Command Descriptions". Refer there for the details of each. The remainder of this section continues the sample walk through of a restore using CART.

Once CART's prompt appears, the user issues the ATTACH command to make a logical connection between CART and the local UNIX tape device. (Think of ATTACH as a replacement for the USER-Access CONNECT command). ATTACH requires, as its only parameter, the name of the local tape device in which the tape is mounted (in this example /dev/rmt0):

```
User> attach /dev/rmt0
User>
```

Upon a successful ATTACH, a new prompt is displayed. In the event that the ATTACH failed, an error message would be printed.

Once attached to the local UNIX tape device, the user restores files in virtually the same manner as done through Central Archiving. One has the option of invoking the LIST or RESTORE aliases, or using the RECEIVE command. The only difference is that the container file to be restored now exists on a local UNIX tape as opposed to a remote IBM tape or DASD. The following abbreviated example uses the RESTORE alias to restore all of the files in the archived container file which spans three consecutive 9-track tapes:

```
User> restore
Container file name? *
File(s) to restore?

Mount tape #1 - hit RETURN when ready:
CART: Tape file ID: UNX1ROOT.JAN2290    Volume sequence number: 0001

x database.inx, 98177 bytes, 61 tape blocks
x dbstat.ua, 86321 bytes, 57 tape blocks

Mount tape #2   hit RETURN when ready:
CART: Tape file ID: UNX1ROOT.JAN2290    Volume sequence number: 0002

x dbstat.wp, 307872 bytes, 210 tape blocks
x document.txt, 107656 bytes, 70 tape blocks

Mount tape #3   hit RETURN when ready:
CART: Tape file ID: UNX1ROOT.JAN2290    Volume sequence number: 0003

x helpfile.txt, 98288 bytes, 61 tape blocks
x login.csh, 110222 bytes, 71 tape blocks
x manual.txt, 80142 bytes, 54 tape blocks
x status.log, 220754 bytes, 143 tape blocks

CART: Source                  Destination                       Size
CART:              ---
CART: *                       !tar xvf -                     1211104
User>
```

A container file name of "*" above tells CART to restore the next container file found on the tape set. If a tape set contains multiple container files, the user has the option of specifying the exact container file name (case is significant), or skipping past a specified number of files to position to the desired one (see the "RECEIVE (RESTORE & LIST) Command Support for CART" command in "CART Command Descriptions"). After prompting for all RESTORE parameters, CART begins the restore process.

By default, CART prompts prior to reading each tape. Once a tape is loaded and online, the user should respond to the prompt with a <RETURN>. CART then reads the tape and reports the "Tape file ID" (or

data set identifier) as recorded in the tape label. In the example above, this is the actual name of the IBM data set or container file UNX1ROOT.JAN2290 (only the last 17 characters are stored). CART also displays the "Volume sequence number" of the tape. This is displayed for information purposes only but not validated in any way by CART.

Once past the tape label, CART verifies that the first portion of the container file contains the Central Archive header information originally stored with the file or data set. If no header is found, or it is unintelligible, CART reports an error, just as Central Archiving does when restoring from an invalid container file. If the container file contains a valid header, CART proceeds to read blocks of data from the tape and passes them directly to the UNIX restore utility being used (in this example, tar). Notice in the example above, CART and tar output are intermixed. This is normal since both processes are simply writing output as information becomes available.

At the end of a tape, CART rewinds the tape by default and prompts for the next one in the sequence. The user should load the next tape, turn the drive online, and hit <RETURN> in response to the prompt. CART automatically picks up where it left off and continues the restore process. When the end of the container file is reached, CART displays the standard USER-Access file transfer status information as shown above. At this point the restore is complete and the tape drive can be detached. The user may then exit CART:

```
User> detach
User> exit
```

The sample session described above used the CART command qualifier defaults, without modification. These defaults can be displayed or modified as done with any USER-Access command using the SHOW or SET commands respectively. The "CART Command Descriptions" section addresses the CART command qualifiers in detail.

Using the CART Utility with CAM

The container file format written by CAM UABACKUP is different from the standard Central Archiving container file format used by USER-Access and by the Central Archiving Management Tools. In addition, the container file created by CAM may span several segments. The first step to using CART with a container produced by CAM is to append all container file segments together on a tape volume set. The following JCL shows an example of how ten segments may be appended to a single tape volume set.

```
//HENSLEDG JOB (ACCT),'H213CAM CART COPY',
//            CLASS=A,MSGCLASS=X,NOTIFY=HENSLED,COND=(0,NE)
//*
//*  REFER: HENSLED.TSO.CNTL(IEBGCART)
//*  COMP:  NSC H213CAM IBM/MVS USER ACCESS CAM
//*  DOC:   COPY A SEGMENTED CAM BACKUP FOR CART USE
//*
//IEBGENER EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=TMD.CAM.DB.UNIX1.TEST.FULL.CCRPCSY.S1,DISP=SHR
//         DD DSN=TMD.CAM.DB.UNIX1.TEST.FULL.CCRPCSY.S2,DISP=SHR
//         DD DSN=TMD.CAM.DB.UNIX1.TEST.FULL.CCRPCSY.S3,DISP=SHR
//         DD DSN=TMD.CAM.DB.UNIX1.TEST.FULL.CCRPCSY.S4,DISP=SHR
//         DD DSN=TMD.CAM.DB.UNIX1.TEST.FULL.CCRPCSY.S5,DISP=SHR
//         DD DSN=TMD.CAM.DB.UNIX1.TEST.FULL.CCRPCSY.S6,DISP=SHR
//         DD DSN=TMD.CAM.DB.UNIX1.TEST.FULL.CCRPCSY.S7,DISP=SHR
//         DD DSN=TMD.CAM.DB.UNIX1.TEST.FULL.CCRPCSY.S8,DISP=SHR
//         DD DSN=TMD.CAM.DB.UNIX1.TEST.FULL.CCRPCSY.S9,DISP=SHR
//         DD DSN=TMD.CAM.DB.UNIX1.TEST.FULL.CCRPCSY.S10,DISP=SHR
//SYSUT2   DD DSN=TMD.CAM.DB.UNIX1.TEST.CART,
//            DISP=(NEW,CATLG,DELETE),
//            DCB=(TMD.CAM.DB.UNIX1.TEST.FULL.CCRPCSY.S1,DEN=3),
//            UNIT=282,LABEL=(1,SL,EXPDT=99000)
//*
//LASTSTEP EXEC PGM=IEFBR14         THIS PUTS COND CODE IN NOTIFY
```

Once the tapes have been created, they can be mounted and CART can be run. The alias UARESTORE can be used to restore from a container file:

```
$ /usr/nsc/sicom/user/cart
User> attach /dev/rmt0
User> uarestore
Container file name? *
File(s) to restore [all]?
Destination [/tmp]? /data/myers/tmp/cartrestore
UARESTORE options [ verbose  replace]?  verbose

Mount tape #1   hit RETURN when ready:
CART: Tape file ID: UNIX1.TEST   Volume sequence number: 0001
Directory /data/myers/tmp/cartrestore/ua_m2
File /data/myers/tmp/cartrestore/ua_m2/install
      .
      .
      .
Directory /data/myers/tmp/cartrestore/sicom/cam
File /data/myers/tmp/cartrestore/sicom/cam/client
File /data/myers/tmp/cartrestore/sicom/cam/clihelp.cam
File /data/myers/tmp/cartrestore/sicom/cam/client.res
File /data/myers/tmp/cartrestore/sicom/cam/CAM
CART: Source              Destination                      Size
CART:
CART: *                   !uarestore /data/myers/tmp/c   9493568
CART:                     artrestore  incl *  verbose
User> exit
```

Installing CART on a Non-USER-Access System

In order to run the CART utility from a UNIX system, it must be installed. This will be the case on a system in which USER-Access is installed since CART is part of the general USER-Access release. If, however, USER-Access is not installed and running on the UNIX system performing the CART restore, a brief installation must first be performed.

The installation of CART is simply a modified version of a USER-Access install as described in the USER-Access MEMO TO USERS provided with the product. For the CART install, this MEMO TO USERS should be in-hand. The steps referenced below refer to the steps described in "Installation Procedure". The installation prerequisites also must be met.

Step 1. Create the USER-Access DISTRIBUTION Directory

Follow this step exactly as described.

Step 2. Load the Distribution

Follow this step exactly as described.

Step 3. Install the Distribution

All that is required here is to install the CART executable. This can be accomplished by typing:

```
$ ./install
```

Select "CART" as the installation option, and answer the remaining questions.

Step 4. Update Network Databases

Skip this step completely.

Step 5. Start the Service Initiator

Skip this step completely.

Step 6. Verify USER-Access

This step should be ignored. Instead, verify that CART can be invoked by typing:

```
$ /usr/nsc/sicom/user/cart
```

A prompt should appear, similar to "User>". Type EXIT in response to the prompt.

At this point CART is installed. Refer back to "Running the CART Utility" for an example walk through.

Limitations and Restrictions of CART

The following are limitations and restrictions of the CART utility.

- CART does no validation of the VOLUME SEQUENCE NUMBER found in the tape label. It is up to the operator to verify tape sequence. CART does, however, display this information as each new tape is read. This restriction should have no damaging affect on a file system since the restore utilities themselves will reject data that it encounters out of order.

- CART supports the following tape record formats:
  - Type U - undefined length records (IBM, ANSI)
  - Type F - fixed length records (IBM, ANSI)
  - Type V - variable length records (IBM)
  - Type D - variable length records (ANSI)

CART has no support for records that span blocks.

- CART does not support BLOCK PREFIX for FIXED or ANSI_VAR records.
- CART supports a maximum tape block size of 32767.

CART Command Descriptions

This section describes the new CART commands ATTACH and DETACH, as well as the RECEIVE command as it relates to the CART utility. Each of these commands function as standard USER-Access commands.

ATTACH Command

Description

The ATTACH command is used to attach the CART utility to a local system tape device. An ATTACH must be performed prior to restoring data through CART. Each new ATTACH does an implied DETACH (if a tape device was previously attached), which causes the tape to be rewound. If the value of qualifier ONLINE is OFF, the tape will be taken offline following each ATTACH or rewind.

Format

```
ATTach <qualifiers> tape_device
```

Where:

<qualifiers>	An optional list of valid command options as described below.
tape_device	Name of a local system tape device upon which a tape containing the container file to be restored is mounted.

Qualifiers

BLOCKsize	The maximum tape block size given in bytes. The valid range for this value is 0 to 65534. The default is 32767.
LABel	Type of tape label recorded on the tape to be read. Valid types are IBM (IBM labeled tape), ANSI (ANSI labeled tape), NONE (tape was written with no label), and STREAM (no label processing or file is not a tape device). CART uses this value to decipher the information stored on the tape. The default is IBM.
ONLine	A Boolean value specifying whether or not to leave the tape online following a rewind. Valid values are ON and OFF. The default is OFF, which says take the tape offline following a rewind and at each DETACH command. This value has no affect when the PROMPT qualifier is OFF, at which time the tape device is always taken offline by CART following a rewind.

PROMpt — A Boolean value specifying whether or not to prompt the user (operator) following a new tape mount. Valid values are ON and OFF. A value of ON forces the user to hit <RETURN> following a new mount. A value of OFF tells CART to begin reading the next tape as soon as the tape device is turned online. OFF is suggested when tape mounts are being done in a separate room by an operator not running CART.

VOLume — A string containing the volume serial number of the tape to be read. If a value is specified, CART verifies each tape of the tape set, making sure its label contains the specified volume serial number. Each tape must have the same one. The default is no VOLUME value, in which case CART ignores the volume serial number on each tape label.

Examples

To attach to the local tape device which contains a tape with an ANSI label, issue the following:

```
User> attach -label ansi /dev/rmt0
User>
```

The fact that no error message was returned says the ATTACH was successful.

A value of STREAM for the ATTACH LABEL qualifier tells CART to process a stream of data with no tape label processing. This allows CART to operate on data where the operating system performs the tape label processing. CART can also read disk files or piped data using the STREAM tape label option.

The following example shows how CART could be used to restore from a CAM UABACKUP container file that exists on an available filesystem. The "file" in this example could also be a named pipe. The "uarestore" alias is used to perform the restore. Note that a false name must be given when prompted for "Container file name" (e.g., "bogus_name" is given). The real name is given in the ATTACH command (e.g., "my_container"):

```
$ /usr/nsc/sicom/user/cart  rootdir /usr/nsc/sicom/user/
User> attach my_container -label stream
User> uarestore
Container file name? bogus_name
File(s) to restore [all]?
Destination [/tmp]?
UARESTORE options [ verbose  replace]?
```

To set the ATTACH qualifier ONLINE to a default of ON, and qualifier BLOCKSIZE to a default of 8192, issue the following commands:

```
User> set attach online on
User> set attach blocksize 8192
User> show attach
CART:
CART:   BLOCKsize ......... 8192
CART:   LABel ............. IBM
CART:   ONLine ............ on
CART:   PROMpt ............ on
CART:   VOLume ............
CART:
User> attach /dev/rmt0
User>
```

The SHOW ATTACH lists the new default qualifier values. The following ATTACH command above, now uses the new values.

DETACH Command

Description

The detach command causes CART to detach itself from a previously attached local tape device. DETACH causes the currently mounted tape to be rewound. An EXIT or QUIT from CART does an implied DETACH.

Format

```
DETach
```

There are no qualifiers or parameters for the DETACH command.

Example

Assume the user is attached to a local tape device from a previous ATTACH command. To force a detach from that device, the following command is issued:

```
User> detach
User>
```

The DETACH command above also would cause the tape to be rewound.

RECEIVE (RESTORE & LIST) Command Support for CART

Description

The RECEIVE command is used to read data from a previously attached tape device. Generally the data is read directly into a local system restore utility. The LIST and RESTORE aliases also use RECEIVE. When possible, LIST and RESTORE are suggested for use with CART instead of a direct RECEIVE command.

Format

```
RECeive <qualifier> source destination
```

Where:

<qualifiers>	An optional list of valid command options for the RECEIVE command. Those new for the CART utility are described below.
source	The source container file name from which to restore. For single file tape sets, this can easily be specified as just an asterisk '*'. For multiple file tape sets, an asterisk can be specified (to read the next container file from the tape), or the exact container file name to read may be specified. When specifying a file name, the character case is significant. Usually uppercase is expected. File names that contain embedded blanks must be enclosed in double quotes. CART uses only the rightmost seventeen characters of any file name specified since tape labels hold only that many file name characters.
destination	The destination file name. For CART, this should be a string containing an invocation of the local restore utility which is to receive the source container file. Examples of such a string appear in the definitions for aliases LIST and RESTORE.

Qualifiers

MODe	For the CART utility, valid values for MODE are RESTORE and STREAM only. The default is RESTORE.

REWind — A Boolean value indicating whether or not to rewind the tape at the end of a successful transfer from a container file. A value of OFF causes the tape to be left positioned at the next container file on the tape. The default value of ON forces the tape to be rewound after reading the selected container file.

SKIp — A numeric value indicating the number of tape files in which to skip prior to reading. This qualifier is used for multiple file tape sets. For labeled tape (IBM or ANSI), logical files are skipped. For unlabeled tape (NONE), SKIP refers to the number of physical tape marks to be skipped. Using SKIP, a user can skip over unwanted container files to get to the desired one. In order to use SKIP effectively, the user should know the order in which container files appear on the tape. For labeled tape, skipped file names are displayed. The valid range for this qualifier is 0 to 32767. The default is 0.

Examples

Assuming a multiple file tape is mounted and attached to a local tape device, the following command lists out the contents of the third container file:

```
User> list * -skip 2
Mount tape #1 - hit RETURN when ready:
CART: Tape file ID: LIBRARY.BCK       Volume sequence number: 0001
CART: Tape file ID: DOCUMENTS.BCK     Volume sequence number: 0001
CART: Tape file ID: TEST.BCK          Volume sequence number: 0001

rw-rw---- 10/2   3177 Jan 18 13:55 1990 cap.c
rw-rw---- 10/2   1632 Jan 18 13:55 1990 cap.h
rw-rw---- 10/2  12532 Jan 18 13:55 1990 cmd.c
rw-rw---- 10/2   2887 Jan 18 13:55 1990 cmd.h
rw-rw---- 10/2  17222 Jan 18 13:55 1990 env.c
rw-rw---- 10/2   7713 Jan 18 13:55 1990 env.h
rw-rw---- 10/2  66321 Jan 18 13:55 1990 file.c
rw-rw---- 10/2  10029 Jan 18 13:55 1990 file.h

CART: Source                   Destination                 Size
CART: ------------------------ ------------------------ -------
CART: *                        !tar tvf -                 118624
User>
```

Notice that the first two container files on the tape were skipped.

To specify a particular container file from a tape set, give the exact container file name. For example, to restore all of the files stored in container file MYDATA.BCK on the attached tape, issue the following RESTORE command. Prior to the RESTORE, specify that the tape not be rewound following the transfer by setting the RECEIVE qualifier REWIND to OFF:

```
User> set receive rewind off
User> restore
Container file name? MYDATA.BCK
File(s) to restore?

Mount tape #1 and place ONLINE when ready.
CART: Tape file ID: SYSTEM.ONE.BCK     Volume sequence number: 0001
CART: Tape file ID: SYSTEM.TWO.BCK     Volume sequence number: 0001
CART: Tape file ID: MYDATA.BCK         Volume sequence number: 0001
```

The remainder of the output is not shown but the resulting container file MYDATA.BCK would be restored, with the tape left positioned at the next container file if one existed.

CART SPECIFIC ERROR MESSAGES

FAC	CODE	SEV	COM	TEXT
UA	5902	E		Invalid SSS tape label [SSS]
UA	5903	E		Bad volume serial number [SSS]
UA	5904	E		There is no tape device attached
UA	5905	E		Invalid variable block size NNN
UA	5906	E		Block Descriptor Word count NNN is invalid (NNN)
UA	5907	E		Record Descriptor Word count NNN is invalid (NNN)
UA	5908	E		Record length NNN is too large for buffer NNN
UA	5909	E		Invalid tape record format SSS
UA	5910	E		Failed to skip NNN tape files
UA	5911	E		Failed to find tape file ID: SSS
UA	5912	E		Double tape mark encountered
UA	5913	E		Missing header label (HDR1)
UAxx3	9101	E	H	Failed to REWIND the magnetic tape

Appendix A: Updating the TCP/IP network control files

For TCP/IP:

The /etc/services network control file can be updated to change the USER-Access default. If Network Information Service (NIS - formerly Yellow Pages) is active, update the equivalent files on the master NIS server and 'make' the changes.

In the /etc/services file, add a TCP/IP 'user' service. You must first select a unique port number. Port numbers in the range 0 5000 are reserved for TCP network services. USER-Access uses port number 6900 as the default. For example, add the following line to the end of the file:

```
user        6900/tcp    USER
```

For USER-Gate:

When using the USER-Gate protocol, many references are made to the Gateway node and to Satellite nodes. The USER-Gate software is a product which uses TCP/IP to provide a gateway to the NETEX backbone for hosts which have no direct NETEX connection. The node which has NETEX and TCP/IP installed is termed the Gateway node (typically a UNIX system). The other hosts (PC's, fileservers, UNIX systems, etc.) which connect to NETEX via the TCP/IP gateway are termed Satellite nodes.

The /etc/services network control file can be updated to change the USER-Access default. If Network Information Service (NIS - formerly Yellow Pages) is active, update the equivalent files on the master NIS server and 'make' the changes.

In the /etc/services file, add a TCP/IP 'usergate' service. You must first select a unique port number. Generally, you should choose a port number above the range 0-5000 (reserved Internet port numbers). Each host installing USER-Access and USER-Gate at your site must select the same port number for the 'usergate' service. If this is the first USER-Access/USER Gate install at your site, a port number of 6930 is recommended, assuming it is not being used by any other TCP/IP application.

Note: If yours is an existing USER-Access/USER-Gate site, you may be using the previous recommended port number of 3000. You should continue to use the port number that was originally selected.

Note: The port number must agree with the port on which the 'usergate' service is offered on the gateway host. Refer to the appropriate network control files on the gateway host. For example, add the following line to the end of the /etc/services file, replacing '####' below with the port number you have selected (e.g., 6930, 3000):

```
usergate    ####/tcp      USERGATE
```