

sdStorTM

Document Storage and Retrieval
Version 1.0

©2003 by Synergetic Data Systems Inc. All rights reserved.
sdStor is a trademark of Synergetic Data Systems Inc.

UnForm is a registered trademark of Synergetic Data Systems Inc. Other product names used herein may be trademarks or registered trademarks of their respective owners.

The sdStor software is ©2003 by Allen Miglore. All rights reserved.
Published under license by Synergetic Data Systems Inc.

Table of Contents

| | |
|---|-----------|
| TABLE OF CONTENTS | 2 |
| INTRODUCTION | 4 |
| INSTALLATION | 5 |
| Requirements..... | 5 |
| Unix CD Installation..... | 6 |
| Unix Download Installation | 6 |
| Windows Installation..... | 7 |
| CONFIGURATION | 9 |
| The sdstor.ini File | 9 |
| User .sdstor Profiles | 9 |
| The sdstord.ini File | 10 |
| The sdstorw.ini File | 11 |
| LICENSING | 12 |
| COMMAND LINE OPTIONS | 14 |
| CAPTURING AND STORING REPORTS | 18 |
| Archive Libraries..... | 19 |
| Configuration in BBx..... | 20 |
| Configuration in ProvideX..... | 21 |
| Configuring Server Printing | 21 |
| VIEW SERVER | 23 |
| COMMAND LINE REPORT RETRIEVAL | 25 |
| Scanning Libraries..... | 25 |
| Retrieving Reports..... | 26 |
| SDSTOR VIEWER | 28 |
| Logging In | 28 |
| Search Form..... | 29 |
| Scanning Window..... | 31 |
| Report View Windows | 32 |
| Report Properties | 34 |
| Administration Tools..... | 34 |
| AUTOMATING REPORT PROPERTIES WITH RULE FILES | 36 |
| APPDATE | 37 |
| APPWHO | 38 |
| CODE BLOCKS – PAGE{}, JOB{}..... | 39 |
| CONST..... | 42 |
| DATEFMT | 43 |
| DETECT | 44 |
| DOCID..... | 45 |
| KEYWORDS | 46 |
| KWSEP | 47 |
| LIBRARY | 48 |
| MEMO | 49 |
| MERGE | 50 |

| | |
|---|-----------|
| PAGE | 51 |
| TITLE | 52 |
| RULE FILE EXAMPLES..... | 53 |
| Invoice Rule Set – sample1.txt..... | 54 |
| Statement Rule Set – sample2.txt..... | 55 |
| AR Aging Rule Set – sample3.txt | 56 |
| Purchase Orders Ruleset – sample4.txt..... | 57 |
| COMMAND LINE ADMINISTRATION..... | 58 |
| SECURITY..... | 62 |
| REGULAR EXPRESSIONS | 64 |
| COMMAND LINE TUTORIAL..... | 65 |
| Change admin password..... | 65 |
| Add a regular user..... | 65 |
| Add sample library..... | 65 |
| Add read rights for sample library to the regular user..... | 65 |
| Capture reports to the sample library | 66 |
| Scan the library | 66 |
| Extract a report from the archive | 67 |
| Using rule files for automatic capture options..... | 68 |
| Deleting an archive from the sample library | 68 |
| INTEGRATING WITH UNIFORM | 70 |
| Configuration..... | 70 |
| Command Line Control | 70 |
| Aliased Printer/Link File Control | 71 |

Introduction

sdStor is a report archiving and retrieval software product, designed to accept text reports, store them in an archive library, and pass them on to their ultimate print device. sdStor consists of three elements, two of which reside on the server where the reports are printed, and one of which resides on Windows workstations throughout your network.

Capture program

The first element is the capture program. It is normally integrated into the printing environment for your application. It accepts printed data, stores it, and passes it on. The capture program can store reports using default information for the report title, date, and other information. Report information can be updated later to refine the title, add keywords and comments and more. The capture program is also designed to work with a set of rules that define how to manage the archive for any particular report. By defining rules for the capture program, you can pre-determine many data elements associated with a report in order to make retrieval easier and more efficient. These data elements include the report title, application date and user, memo, and keywords.

There can be any number of libraries on the system, and the capture program can be instructed in several ways which library to store a given report in.

The capture program can also be executed as a command line with options to scan or retrieve reports for reprinting or analysis, or to transfer reports between libraries, or to delete unwanted reports. Appropriate security is required for all these operations.

View Server

The second element is a View Server that handles requests from Windows workstations to scan and retrieve reports. It uses tcp/ip sockets to communicate with the client software.

Viewer

The third element is the Microsoft Windows-based sdStor Viewer software, which provides access, via the background server, to the libraries of reports. This software can scan libraries looking for reports, and view and/or reprint reports.

Assuming appropriate permissions, users can also transfer reports between libraries, and delete reports, using either the capture program command line or the Windows software.

Installation

Installation of sdStor varies between Unix and Windows, between CD and download, and between bundled and unbundled versions. The primary goals in all cases are to get sdStor files loaded into a directory, get the product linked to a run-time engine, and generate a launch script or executable.

It is important to know up front whether you are installing a bundled or unbundled version. This determines which download you retrieve from the SDSI website, or how you answer the initial questions when installing from CD. The difference between bundled and unbundled versions is whether sdStor includes a run-time engine. This run-time can be either Best Software's ProvideX, or Basis International's PRO/5 (Visual PRO/5 on Windows), revision 2.2 or higher. If you already have one of those products, then you need to install the unbundled version. If not, you need to install a bundled version, which includes a run-time PRO/5.

Note that in addition to installing this sdStor server product, you will also want to install the sdStor Viewer software on each Windows workstation that will be accessing the archive libraries. This is stored in a separate CD directory and download file.

Requirements

Disk space:

Library archives can grow large – a million report pages could take 2GB or more. While libraries can be configured to be stored on any file system, there are two libraries that are always placed in the sdStor directory: the default library and the trash library. Without administrative management of libraries, these files can grow very large, requiring that sdStor be placed on a file system with plenty of disk space.

Gzip compression:

Report data is compressed using gzip or compress. The gzip version is preferable, as it provides greater compression. The Windows version includes a copy of gzip.exe. For Unix versions, you should verify that gzip is present on your system by entering **gzip -h** at the Unix command line. If gzip is available, you should see a list of options rather than an error message. If you do not have gzip, then you can obtain an executable for your system from <http://gzip.org>. Install it in a universal executable directory, such as /usr/bin.

Microsoft Internet Explorer and Adobe Acrobat Reader

The sdStor Viewer requires both these programs to display reports. All Windows systems include Internet Explorer, and most already have the Acrobat Reader (or the full version of Acrobat). However, if you need to install Acrobat Reader, you can install it from the CD or download the latest version from <http://adobe.com>. Acrobat should be installed as a plug-in for your web browser, which is its default state when installed after Internet Explorer. In addition, you may wish to turn off the splash screen using the Preferences window in Acrobat.

Unix CD Installation

1. Login as root.
2. Mount the CD as a file system that supports lowercase file names. If you are unsure how to do this, check your man pages: **man mount**. The following table illustrates sample mount commands for various operating systems, assuming the mount directory, /mnt, is available, and standard CD device names. You may need to adjust these commands according to your configuration.

| | |
|--------------|---|
| SCO UNIX OS5 | mount -o lower /dev/cd0 /mnt |
| SCO UNIX | mount -r -f HS,lower /dev/cd0 /mnt |
| UNIXware | mount -F cdfs -r /dev/cdrom/cdrom1 /mnt |
| AIX | mount -v cdrfs -r /dev/cd0 /mnt |
| Sun Solaris | mount -rt hsfs /dev/sr0 /mnt |
| HP/UX | mount -r -F cdfs -o cdcase /dev/dsk/c1d0s2 /mnt |

3. Change to the sdStor UNIX directory in the mount directory: **cd /mnt/sdstor10/unix**
4. Run the install script: **./install.sh**, or if you do not have execute permission to the file, **sh install.sh**. Follow the prompts to select a directory for sdStor, and choose if you will be supplying your own run-time PRO5 2.2+, or PVX engine, or installing a bundled version that includes a pro5 engine. If you install a bundled version, you will need to select the version that is appropriate for your operating system.
5. sdStor will then be installed by copying files to the selected directory, and executing the set up script **./setup.sh** in the install directory. Once sdStor has been installed from the CD, **./setup.sh** can be executed again at any time from the sdStor directory. See step 4 in the next section for more information.
6. Use the **sdstor10 -v** command to ensure sdStor is installed and set up correctly. The output from this command will display the serial number used and state that this is a demo version.
7. If you have already purchased a license, see the **Licensing** section for activation instructions.

Unix Download Installation

1. Login as root.

2. Create a directory to hold the sdStor files, and change to that directory.

Example: **umask 0**
 mkdir /usr/sdstor10
 cd /usr/sdstor10

3. Uncompress and extract sdStor from the download file:

uncompress sdstor10_xxx_tar.Z
 tar xvf sdstor10_xxx_tar

4. Execute the setup.sh script, which may ask you where the PRO/5 2.2+ or ProvideX executable is located on your system. SdStor uses PRO/5 or ProvideX as a run-time environment. A PRO/5 runtime must be revision 2.2 or higher. If you have a bundled version of sdStor that includes a run-time, then this question won't be asked.

./setup.sh

If you see the question "What directory contains PRO5 2.2+ or PVX?", then you must answer with the correct directory, or the installation will not complete, and you will have to re-run this step again when you know the correct directory information. Note that if you use a PRO/5 revision prior to 2.2, sdStor will not run. If your application runs under an older revision, then you can license a bundled version.

The script will then prompt for a default rule file, initially defaulting sdstor.rul. This will be used if the command line doesn't contain a "-f" argument.

Once a valid directory has been entered, setup.sh will create scripts called sdstor and sdstord in the /usr/bin directory. The sdstor script is the command line interface and capture program. The sdstord script launches the view server daemon.

5. Enter **sdstor10 -v** to ensure sdStor is installed and set up correctly. The output from this command will display the serial number used and state that this is a demo version.
6. If you have already purchased a license, see the **Licensing** section for activation instructions.

Windows Installation

1. From the CD, use explorer to locate the D:\sdstor\win directory (or D:\sdstor\winbun for a bundled install that includes a run-time), and double-click the setup.exe program. If you downloaded sdStor from the Internet, simply execute the downloaded executable. Follow the on-screen prompts from the installer to install sdStor to your system. Note that the sdstor.exe and sdstord.exe programs and the associated .ini files are installed both in the Windows directory and in the UnForm directory.
2. Click the **Run setup** icon. This will conditionally rename certain files and prompt for where your run-time Visual PRO/5 or ProvideX is located. The full path to the executable must be given. Use the Browse button if desired to locate the executable. If you installed a bundled version, the selection of the runtime c:\sdsi\rt\vpro5.exe will automatically be made. It will also prompt for a default rule file, to be used in case the sdStor.exe command line doesn't contain a "-f" argument.
3. Use the **sdStor Information** icon to ensure sdStor is installed and set up correctly. The output from this command will display the serial number used and state that this is a demo version.
4. If you already purchased a license, see the **Licensing** section for activation instructions.

Configuration

There are two configuration files used by the two sdStor server elements. These files are called `sdstor.ini` and `ststord.ini`.

The `sdstor.ini` File

The `sdstor.ini` file contains configuration settings for the command line interface, divided into three sections: `[defaults]`, `[uniform]`, and `[paper]`.

The defaults section can contain the following lines:

| | |
|----------------------------------|--|
| <code>century=nnnn</code> | The default century used when dates are provided with 2-digit years. This should generally be set to 2000. |
| <code>datefmt=mdy dmy ymd</code> | The default date format if not specified in a rule set, on the command line, or in the user's <code>.sdstor</code> profile. |
| <code>ff=0 1</code> | Set to 1 to cause a trailing formfeed to be appended to text and laser archive retrievals. |
| <code>login=login</code> | The default login ID used if no ID is provided on the command line or in a user's <code>.sdstor</code> profile. |
| <code>page=n</code> | Sets the default page size if no formfeed is found in a job stream. This value can be overridden on the command line or in a rule set. |
| <code>pswd=pswd</code> | The default login password if no password is provided on the command line or in a user's <code>.sdstor</code> profile. |
| <code>waitonexit=n</code> | Sets the number of seconds to pause before exiting. This may be necessary in some installations of ProvideX to avoid timeout errors. |

The uniform section can contain the following lines:

| | |
|------------------------------|---|
| <code>command=command</code> | The UnForm command for your system, usually <code>uniform50</code> . |
| <code>dfrule=rulefile</code> | The default rule file if a null rule file is specified in the command line. |

User `.sdstor` Profiles

Default information for login, password, and date format can be specified in a user's `.sdstor` profile. This file is named ".sdstor" by default, in the directory specified by the `$HOME` environment variable. It can also be specified as a full path in the `$SSDSTORDEFS` environment variable. This file should be owned by the login user and given read/write owner permission and no non-owner permissions (`chmod 600`). This file

can also specify the user's default library (deflib= line) and date format (datefmt=mdy|dmy|ymd line). An example user .sdstor profile might look like this:

```
# John Smith's .sdstor profile
login=johns
pswd=John'sDogSam
deflib=ar_reports
datefmt=mdy
```

The sdstord.ini File

The sdstord.ini file controls many operational parameters of the View Server. This server is a tcp/ip server that listens on a primary port and requires access to other process ports as well. It can perform logging and can limit access to specific client IP addresses or address wildcards.

The defaults section contains the following entries:

| | |
|-------------------------|--|
| port= <i>port</i> | The primary listening port, which must be an available port between 1024 and 65535. The default port is 5919. |
| logfile= <i>logfile</i> | A log file name for simple or detailed logging of connections. |
| logdetail=0 1 | Set to 0 for simple connection logging, or 1 for detailed logging. |
| timeout= <i>seconds</i> | The number of seconds of inactivity before disconnecting. |
| uniform=0 1 | Allow UnForm-enhanced PDF documents to be generated for the Viewer. The settings for UnForm execution are in the sdstor.ini file. Be sure that the UnForm rule file is properly designed to support PDF documents. |

The security section contains the following entries:

| | |
|----------------------------|---|
| allow= <i>ip1;ip2...</i> | A list of IP addresses or address wildcards that are allowed to access the server. Delimit with commas or semicolons. |
| procports= <i>portlist</i> | <p>A list of ports or port ranges to use when launching processing tasks, each of which listens for the duration of a connection on a unique port. Delimit with commas or semicolons. List port numbers or ranges in the form of <i>start-end</i>, such as 9000-9100. If you don't set a value, then random ports will be used. However, if you have a firewall between clients and the server, it may be necessary to specify a range of ports here and in the firewall configuration.</p> <p>Note that the number of procports provided should be at least twice the number of sdStor Viewer seats licensed. One procport is used for each login, plus one for each scan window opened.</p> |

| | |
|--------------|--|
| nodelete=0 1 | The Windows clients can perform delete tasks on the archive libraries. If you would prefer that only command line deletions can be performed, set this value to 1. |
| noemail=0 1 | The Windows client can email reports as pdf files via a menu option. If you would prefer that emailing is not available in clients, set this value to 1. |

The sdstorw.ini File

The sdstorw.ini file is used exclusively in Windows installations, by both the capture program sdstor10.exe and the view server sdstord10.exe. This file is generally created and maintained by the setup option of the sdStor Start menu group on the Windows system where sdStor is installed. It contains the information about how to start the runtime VPRO/5 or ProvideX.

These are the entries for the [runtime] section of the file.

| | |
|----------------------------|--|
| bbexec= <i>runtime exe</i> | The full path to the Visual PRO/5 or ProvideX runtime, such as c:\sdsi\rt\vpro5.exe. |
| bbterm=TINV | Set to a minimized or invisible syswindow terminal alias, when running in a Visual PRO/5 environment. It is added to a -t VPRO/5 command line option. This should be set to TINV. |
| user=sdstor | A user code, which is added to a -u VPRO/5 command line option. This may be necessary if the sdStor capture program will need access to a Basis data server. Otherwise, it is generally ignored. |
| mem=2048 | The memory space to allocate to a running sdStor process. No sdStor functions require a great deal of memory, so 2048 is generally plenty. This value is added to the -m VPRO/5 command line option. |
| stohome= <i>homepath</i> | The sdStor home directory path. |
| dfrule= <i>rulefile</i> | The default rule file for the sdStor capture program. This value is used if no -f option is specified on the sdstor10.exe command line. |

Licensing

When initially installed, sdStor operates in demo mode for up to 30 days. When in demo mode, words in report retrievals and scan listings are randomly substituted with the string *Demo. Note that captured reports are “clean” – that is, stored data in libraries do not contain *Demo strings. It is only retrieval operations that are affected by the demo mode. Once sdStor is licensed, the *Demo substitutions will stop.

sdStor licensing is based upon the runtime serial number and the number of concurrent sdStor Viewer seats desired. When purchasing a license, you must provide the runtime serial number (in the case of bundled licenses, you will be assigned a serial number), and the number of Viewer seats desired. The following seat counts are available: 1, 3, 5, 10, 15, 20, 25, 30, 40, 50, 75, 100, and unlimited.

sdStor uses several “user slots” from the runtime. These slots are shared among all applications that use the runtime, which can include sdStor, UnForm, and your application software. sdStor uses user slots in the following manner:

- The View Server uses one slot as long as it is running, which generally will be all the time.
- Each execution of the capture program uses one slot for the duration of the capture. This is generally very fast, unless a report has a very large number of pages, or takes a long time to produce while a pipe is opened to sdstor10.
- Any time an UnForm process is started to format a retrieved report, it takes a user slot. An UnForm rule set designed to execute nested UnForm jobs will take additional slots as required.

Runtime Licensing

If you purchased a bundled license, then it will include a runtime PRO/5. Note that if you already have an UnForm bundled license, you must use the same runtime license for both products. You will likely need to increase the user count on that runtime, so will need to follow normal licensing steps to get a new license file with a new user count.

To license a bundled runtime you must generate a license request file, using license.sh on Unix or running License Runtime on Windows. That license request includes a serial number, an authorization number, a system ID, and a return email address. Email that request file to licreq@synergetic-data.com, or post the above information in the Runtime Licensing form at <http://www.synergetic-data.com>, and you will receive by return email a license file. That file must be saved or copied into the runtime directory pro5.lic file, replacing any existing pro5.lic file. The runtime directory defaults to /usr/lib/sdsi/rt on Unix, or c:\sdsi\rt on Windows. On Unix, then use license.sh to re-read the license file.

Once the runtime is licensed, you can then activation sdStor.

Activating sdStor

To activate sdStor, enter **sdstor10 -act** on Unix, or run Activate sdStor on Windows. You will be shown your serial number and prompted for your activation key. The serial number shown must match the serial number used to generate the activation key. Enter the key. If no error message is shown, then the product is activated. You can verify this by entering **sdstor10 -v** on Unix, or running sdStor Information on Windows. You should see a line indicating “Live since *date*”.

Command Line Options

The `sdstor10` executable provides numerous command line options. Options are meaningful based on the use of the command. Some commands trigger action, such as `-id` triggering a report retrieval, `-scan` triggering library lists, and `-act` triggering a prompt for an activation key.

Each `sdstor10` command execution uses a single task slot from the run time license until the task is complete. If a retrieval command is executed and sent to a printer configured to use UnForm, then the UnForm execution uses an additional slot.

| | |
|------------------------------------|---|
| <code>-act</code> | Invokes a prompt for an activation key. |
| <code>-appdate <i>date</i></code> | Sets the application report data to <i>date</i> . Dates are entered as delimited digits in month-day-year order, unless a different order is specified with the <code>-datefmt</code> option. The delimiter can be any non-digit character. Examples: 1/31/2002, 1-31-2002. |
| <code>-appwho <i>who</i></code> | Sets the application report user to <i>who</i> . |
| <code>-datefmt <i>fmt</i></code> | Sets the date format to <code>mdy</code> , <code>dmy</code> , or <code>ymd</code> . Date formats determine the order in which date digits are interpreted. |
| <code>-delete</code> | Removes the report specified with the <code>-id</code> option from the library specified with the <code>-lib</code> option (or the default library). The login must have delete rights to the library. If the library is not the trash library, the report is transferred there. Otherwise, it is removed from the trash library. |
| <code>-e <i>file</i></code> | Sets error log file to <i>file</i> . This is where runtime errors and warnings are printed. Without an error log file, runtime errors are printed to standard output on Unix or a message box on Windows. Warnings, such as syntax errors in code blocks, are only printed to an error log file. |
| <code>-f <i>file</i></code> | Sets rule file to <i>file</i> . |
| <code>-fields <i>list</i></code> | Sets the fields to list with the <code>-scan</code> option. The default setting is "id;title;rundate;runtime;pagecount". A special field, <code>keyword</code> , is prepended to this if the scan order is <code>keyword</code> . Valid field names include <code>id</code> , <code>title</code> , <code>rundate</code> , <code>runtime</code> , <code>pagecount</code> , <code>who</code> , <code>appdate</code> , <code>appwho</code> , <code>checksum</code> , <code>memo</code> , <code>maxcol</code> , <code>maxrow</code> , and <code>pagecount</code> . Be sure to quote the <i>list</i> value on Unix if it includes semicolons, spaces, or other characters significant to the shell. |
| <code>-filter <i>filter</i></code> | Sets the scan filter to <i>filter</i> when printing a list of reports with the <code>-scan</code> option. The <i>filter</i> can be a literal value or a regular expression prefixed with a tilde. For example, <code>-filter "Smith and Sons"</code> will find scan lines that contain "Smith and Sons" (non-case sensitive). <code>-filter "~02/[0-9][0-9]/2003"</code> will find |

| | |
|--|---|
| | <p>scan lines that contain the regular expression pattern specified, in this case a date in the month of February, 2003. Filters apply to the print data produced by the scan command, and so can be affected by the <code>-fields</code> option.</p> <p>Be sure to quote the <i>filter</i> value on Unix if it includes semicolons, spaces, or other characters significant to the shell.</p> |
| <code>-i file</code> | Sets input data stream to <i>file</i> . |
| <code>-id id</code> | <p>Retrieves the report archive <i>id</i> from storage. The <i>id</i> must exist in the library specified with the <code>-lib</code> option, or the default library. Generally, this option is used with a <code>-o</code> option to print the archive to a file or device rather than the screen, but on Unix, output may be piped to another program or the spooler. You must have read rights to the library.</p> <p>The format of the retrieved output defaults to plain text, but you can produce scaled pcl6 (LaserJet) format or pdf format using the <code>-laser</code> or <code>-pdf</code> options. When producing one of these formats, each page is scaled to fit the page using actual columns and rows, with minimum values of 80 and 60, respectively. See the <code>-landscape</code> and <code>-paper</code> options as well.</p> |
| <code>-ix file</code> | Sets input data stream to <i>file</i> , and removes the file when done. |
| <code>-keywords keywords</code> | Sets the report keywords to a delimited list of values. The default delimiter is a semicolon, but it may be set with the <code>-kwsep</code> option. |
| <code>-kwsep val</code> | Sets the keyword delimiter to <i>val</i> . The default value is a semicolon. |
| <code>-landscape</code> | Sets the default output orientation for pdf or laser output to landscape. |
| <code>-laser</code> or <code>-pcl</code> | Sets printed output to pcl5 format. |
| <code>-lib library</code> | Sets the library for archive storage, retrieval, or management to <i>library</i> . |
| <code>-libedit rec</code> | Adds, deletes, or updates a library record. Admin users only. |
| <code>-liblist</code> | Prints a library listing. Admin users only. |
| <code>-login name</code> | Sets the login name to <i>name</i> . See the Security chapter for more information. The <i>name</i> setting can also include a password in the format <i>name/password</i> . |
| <code>-memo memo</code> | Sets the report memo field to <i>memo</i> . |
| <code>-noff</code> | Turns off formfeeds between retrieved pages of output. |
| <code>-o file</code> | Sets the output data stream to <i>file</i> . This can also be a device, such as <code>/dev/lpt0</code> or <code>LPT2</code> , or a printer alias line defined in <code>config.sto</code> (PRO/5) or <code>link file</code> (ProvideX). |
| <code>-p laser</code> or <code>-p pdf</code> | Selects the laser or pdf driver for retrieval output. Equivalent to using the <code>-laser</code> or <code>-pdf</code> options. |
| <code>-page n</code> | Sets the input stream to page lengths of <i>n</i> lines per page. This is necessary if there are no formfeeds in the input data stream |

| | |
|-----------------------|---|
| | and no rule set invoked that sets lines per page with the page command. |
| -pages <i>pages</i> | Sets the list of pages to retrieve when used with the <code>-id</code> option. The format for <i>pages</i> is a comma-delimited list of page numbers or ranges, such as “1,3,5-10”. Values greater than the last page on file will print the last page, so it is easy to print the last page with an artificially high number, like “-pages 9999”. |
| -paper <i>paper</i> | Sets the paper size for pdf or laser output to <i>paper</i> . This can be letter, legal, ledger, executive, a4, a3, or any other setting defined in the <code>sdstor.ini</code> file. |
| -password <i>pswd</i> | Sets the login password to <i>pswd</i> . You can also set the password as a suffix to the login <i>name</i> . See the Security chapter for more information about login and password control. |
| -pcl or -laser | Sets printed output to pcl5 format. |
| -pdf | Sets printed output to pdf format. |
| -prm <i>defs</i> | Sets one or more global strings (<code>stbl()</code> in PRO/5 or <code>gbl()</code> in ProvideX) to values that can be referenced in code blocks or expressions in rule sets. The format of <i>defs</i> is “ <i>var1=value1;var2=value2;...</i> ”. |
| -r <i>set</i> | Set the rule set to be invoked to <i>set</i> rather than using detect commands to determine the rule set. The <i>set</i> must be contained in the rule file specified by the <code>-f</code> option. |
| -rebuild | This option is used to rebuild the library data files for the library specified in the <code>-lib</code> option. This will reclaim any available space from deleted records. The library is locked during the process, and users should not attempt to capture or retrieve reports from the library. Be sure to create a temporary backup of the library data files (<i>path.ar1</i> , <i>path.ar2</i> , and <i>path.ar3</i>) in case of failure. Admin users only. |
| -scan <i>settings</i> | Prints a list of reports from the library specified with the <code>-lib</code> option. The login must have read rights to the library. The format of <i>settings</i> is “ <i>order[;start[;end]]</i> ”. The <i>order</i> can be <i>id</i> , <i>title</i> , <i>rundate</i> , <i>who</i> , <i>appdate</i> , <i>appwho</i> , or <i>keyword</i> . There is also a special setting of “keywords” that will just list unique keywords in the library. The <i>start</i> and <i>end</i> values determine the range of values in the order specified. For example, “ <i>rundate;2/1/2003;2/15/2003</i> ” could be used to scan a range in date order from February 1 through February 15, 2003 (assuming <code>-datefmt</code> is <code>mdy</code> , the default). Scan ranges limit the number of records processed, and can therefore be faster than using the <code>-filter</code> option. Be sure to quote the <i>settings</i> value on Unix if it includes semicolons, spaces, or other characters significant to the shell. |

| | |
|--------------------------|---|
| -sep <i>sep</i> | Sets the separator character for listings to <i>sep</i> . The default separator is a semicolon. In addition to characters, <i>sep</i> can be set to tab, lf, cr, or crlf. |
| -title <i>title</i> | Sets the title of this report to <i>title</i> . |
| -transfer <i>lib</i> | Transfers a report specified with the <code>-id</code> option from the library specified with the <code>-lib</code> option (or the default library) to the library <i>lib</i> . The login must have delete rights to the <code>-lib</code> library and write rights to <i>lib</i> . |
| -uniform <i>rulefile</i> | Causes report retrieval to be passed through UnForm for enhanced formatting. If the rule file is null (“”), then the default rule file is taken from the <code>sdstor.ini</code> file. The UnForm command to execute is also found in the <code>sdstor.ini</code> file. UnForm must be configured and operational for this option to work. If enhanced PDF files will be generated, particularly for the sdStor Viewer, the rule file must be properly designed to support PDF output. |
| -useredit <i>rec</i> | Adds, deletes, or updates a user record. Only administrative logins are allowed to use this option. See the Administration chapter for more information. Admin users only. |
| -userlibedit <i>rec</i> | Adds, deletes, or updates a user library record. Admin users only. |
| -userliblist | Prints a user library listing. Admin users only. |
| -userlist | Prints a user listing. Admin users only. |
| -v | Displays version information |
| -x | Prints a crosshair pattern on top of retrieved report output. The crosshair pattern can be produced in pdf or pcl format, but not text format, so you should include <code>-pdf</code> or <code>-laser</code> options. Always used in conjunction with a <code>-id</code> option to retrieve report data, and may be used with the <code>-pages</code> option for overlaying selected pages. This is a helpful design aid for rule set development. |
| -xl <i>pages</i> | Like the <code>-x</code> option, but produces the crosshair pattern output in landscape orientation. |

Capturing and Storing Reports

sdStor is designed to capture text-style reports, using the program `sdstor10`, as they are printed from an application. Once installed into the application's printer, all reports printed to that printer are automatically archived unless specific rules are invoked to disable the archiving process. The archive stores the plain text version of the report as a set of compressed pages. In addition, the following data is stored:

| | |
|-------------------------|---|
| Report ID | A unique, internally generated code for each report. |
| Title | A user-defined report title or name. |
| Run Date | The system date when the report was run. |
| Run Time | The system time when the report was run. |
| Run User | The system login user when the report was run. |
| Application Report Date | A user-defined date. |
| Application Report User | A user-defined user code. |
| Memo | A user-defined comment. |
| Keywords | Zero or more user-defined keywords used for fast searching. |
| Checksum | A system generated value based on report content, which can be helpful in locating duplicate reports. |
| Columns | The widest print column in the report, used for scaling the report when retrieved in pdf or laser format. |
| Rows | The highest print row in the report, used for scaling the report when retrieved in pdf or laser format. |
| Pages | The number of pages in the archive. |

When capturing a report, sdStor automatically sets the report ID, default title, run date, run time, checksum, columns, rows, and pages fields. The command line, or a rule set, can additionally specify a custom title, application date and user, memo, and keywords. The default title is based upon initial report words that make up to 60 bytes.

sdStor reads the input stream, looking for linefeed characters to break lines and formfeed characters to break pages. If no formfeeds are found, then it will break the job up into even numbers of lines per page. The number of lines can be set in a rule file (page command), the command line (-page option), or will default to 66.

Important note: As pages are read, any PCL (laser) escape sequences are removed, leaving plain text lines for storage. Any escape sequences or control characters used by other printer types will probably wind up in the archive data, so care should be taken to print only PCL or plain text through sdStor.

Once the first page is read, the rule file is parsed. If a `-r` option is specified, the named rule set is used. Otherwise, rule set detect commands are compared with the first page's data until a match is found and the detected rule set is used. Rule sets can be used to

automatically set user-controllable fields during the archiving process. For example, a rule set could set the title and a series of keywords, making it easier to locate the report later.

All report data is then processed, with pages compressed and stored, and the raw print data passed through to standard output or the file or device specified with the `-o` option.

Note that if an error occurs while trying to launch the runtime engine that performs the capture process, then no capture occurs. In this case, on Unix installations an error page description will be printed instead of the report to standard output or the `-o` output specification, and an error log file if named with a `-e` option. In addition, you can configure the `sdstor10` script to send an email of the error to a specified address, using the Unix mail command, by setting this line: `ERRMAIL=email address`. In addition to the error message in the `-e` error file and the standard output, the Unix script will also capture the unarchived report to the `lostrpts` directory in the `sdStor` path, reporting the capture file name in the error file, output, and optional email message.

On Windows, an error message will be displayed, and no report output will be printed.

Archive Libraries

Archive data, along with the compressed page data, is stored in a *library*. A library consists of three data files, suffixed by `.ar1`, `.ar2`, and `.ar3`, and is stored anywhere on your system. Administrative users can create libraries using the `-libedit` command or via the `sdStor` Viewer interface.

Using different libraries for your archives provides a way to differentiate application types, such as Accounts Receivable or General Ledger, or report classes, such as Warehouse Reports or Accounting Reports, or dates (2002 General Ledger), or whatever other differentiators you desire. In addition, libraries are an important part of archive security, in that users can be granted read, write, and delete access to specific libraries.

There are two static libraries: `deflib` and `trash`. These libraries cannot be changed or removed. Both reside in the `sdStor` home directory. All other libraries are configurable by an administrative user. See the Administration chapter for more details.

The `deflib` (Default Library) is where all reports that are not assigned a specific library are stored. Reports are assigned to a library by a ruleset, command line (`-lib` option), or a user's default library setting. The rule set takes precedence, followed by the command line, followed by a `deflib` setting in the user's `.sdstor` file, and if no library specification results, then the `deflib` is used.

The `trash` library is used as a holding zone for archives that have been deleted from other libraries. If a report is deleted by accident, it can be transferred back from the `trash` library, until it has been deleted from there.

The three component files of a library store the following data:

- .ar1 Archive data, such as title and dates, for each captured report
- .ar2 Keywords
- .ar3 Compressed pages

When run under ProvideX or PRO/5 revisions less than 3.0, there is a file size limit of 2GB. This means there are limits of approximately 2 million archives and 3 million pages per library. These values are approximate, particularly the page data, as that will vary based upon the average text content per page and how well it compresses. The administrator should periodically check archive sizes, and define new libraries as required.

Configuration in BBx

The most convenient way to archive reports in a BBx application is to modify the config.bbx file's alias lines to include sdStor. The way to do this varies by operating system.

For Unix, where pipes are supported, setup your alias line like this:

```
alias LP "|sdstor10 -f rulefile | lp -dhp4000" "HP 4000 Printer" ...
```

In the above alias, data is piped through sdstor10 on its way to the spooler. The report data is captured by sdStor, and piped on to the lp command.

If you need to integrate sdStor and UnForm in the same printer, be sure to place sdStor first in line, so it doesn't receive binary data from UnForm:

```
alias LP "|sdstor10 -f rulefile1 | uniform50 -f rulefile2 | lp -dhp4000 -oraw" ...
```

For Windows, you must configure the alias line to print to a file and then execute sdStor, like this:

```
alias LP C:/TEMP/LP.TXT "HP 4000 Printer"
O_CREATE,EXECCOFF="C:/SDSI/SDSTOR/SERVER/SDSTOR10.EXE -ix
C:/TEMP/LP.TXT -f rulefile -o LPT2"
```

In the above example, BBx creates the file C:/TEMP/LP.TXT and prints to it. When the printer is closed, it executes c:\sdstor\sdstor.exe with options to read and delete LP.TXT, capture it using definitions in rulefile, and send the output to the device LPT2, which could be a captured or 'net use'd device. Note that you can use backslashes (\), but they must be doubled (\\).

Note that your alias lines will probably contain additional modes for printer initialization, number of columns, etc. What is shown above is just what is required for sdStor-related changes. Be sure that any modes that send escape sequences to the print channel are

configured to send PCL (HP Laserjet) sequences, as those are the only such control sequences recognized and removed by the capture process. Removal of control sequences is required in order for proper text storage and rule set detection.

Configuration in ProvideX

Invoking sdStor within the ProvideX printing environment requires the specification of a custom print driver program in a link file. The print driver “sdstorptr” is copied to your ProvideX *dev directory during setup, and may be used for this purpose. If the default rule file specified during setup is the one you wish to use for any given printer link file AND the printer device is a laser printer (suitable for escape sequences generated by the driver *dev/hplaser), then the stStor driver can be used unmodified. If not, then you can copy the sdStor driver and modify it as appropriate.

Configuring Server Printing

In addition to devices or files, sdStor can also send output to printer aliases defined in the config.sto file. This is useful when the output is going to a UNC device, a Novell spool queue, or if you need to integrate UnForm into the output destined for the printer. Printer aliases in config.sto must start with an uppercase P or L. The lines contain four elements:

- The word “alias”
- A name starting with P or L, followed by letters or digits
- An output device, which on Unix can be a pipe or redirect to a spooling or other operating system command or script. If the device contains spaces, it must be quoted. Embedded quotes, if required, must be escaped with a backslash, as \”.
- A description, which must be quoted if it contains spaces

Here are some examples, assuming sdStor’s command line include “-o P1”:

```
alias P1 “|lp -dlaser1 -oraw -s 2>/dev/null” “Unix Laser Printer”
```

```
alias P1 “|unform50 -f acme.rul | lp -dhp -oraw -s” “UnForm Printer”
```

```
alias P1 //server2/laserjet “Windows UNC alias” spcols=255
```

```
alias P1 nspool “Novell alias” queue=123,spcols=255
```

```
alias P1 c:/temp/p1.txt “UnForm alias”  
o_create,spcols=255,execoff=”c:/unform50/unform50.exe -ix c:/temp/p1.txt -f  
acme.rul -o LPT2”
```

When running under ProvideX, the alias lines in config.sto are read for their names and descriptions. In addition, a link file must be defined in the sdStor directory for each name

specified. The list of aliases named in config.sto is used by the View Server in order to provide a list of valid server printers when the Server Print window is presented in the Viewer. However, when sdStor retrieves a report to a printer named in the -o option, that printer must be a link file.

View Server

The View Server is an “always up” process that listens on a tcp/ip socket for connections from the sdStor Viewer software. The server runs on the same system as the capture program, providing access to the archive libraries for the Viewers.

The server occupies a single task slot from the run-time engine at all times. If clients launch UnForm tasks, then an additional slot is taken by the UnForm task.

Once the server parameters are configured in the sdstord.ini file, the only task is to start it up and keep it running.

Unix Installations

On Unix, the server is managed with the sdstord10 command:

sdstord10 start starts the server.

sdstord10 stop stops the server.

It is probably desirable to start the server when your system boots. To do so, you can edit the /etc/inittab file to add an entry:

```
sdstord10:234:once:/usr/bin/sdstord10 start >/dev/console 2>/dev/console
```

On SCO systems, the inittab file is rebuilt during a kernel link, so you should also add the same entry to the /etc/conf/cf.d/init.base file.

Windows Installations

The sdstord10.exe program is designed to run as a NT service (on NT, 2000, or XP), or as an application. This program in turn manages the runtime that is executing the View Server programs. When run as an application, a window with the View Server log is shown, and you can start and stop the View Server using the Start and Stop buttons. If the process is already installed as a service, then the Start and Stop buttons are disabled, and you just use the Windows Control Panel Services manager to start or stop the process.

The following command line options are provided:

| | |
|----|--|
| /a | Run as an application and view the View Server log. This option is automatically placed on the Start menu as the Server Status option. |
|----|--|

| | |
|------------|--|
| /install | Install the server to run as a service on NT, 2000, or XP. The server will then start automatically at boot time, and can be stopped and restarted from the Windows Control panel Services program. On Windows 2000, the Services program is accessed via the Admin Tools icon on the Control Panel. This option is automatically placed on the Start menu. |
| /uninstall | Uninstall the service. It will show as disabled in the Services program until the next reboot. This option is automatically placed on the Start menu. |
| /start | Start the view server as an application, but do not show the application window. This option is useful if you wish to place a shortcut on the Windows Startup menu, so that the server will execute upon login. If the server is running on a non-NT-based machine, such as Windows 98 or ME, then using the Startup menu is as close as you can get to running the server as a service. |

Command Line Report Retrieval

The sdStor command line provides library scanning and report retrieval capabilities. You must have read rights to a library in order to scan or retrieve a report.

Scanning Libraries

Libraries can contain hundreds or thousands of report archives. In order to find a given report, you can scan a library using the `-scan` option. Specify the library to scan using the `-lib` option, and your login, if necessary, using the `-login` and `-password` options. A number of other options are also recognized when scanning, but the `-scan` option is required.

The `-scan` option requires a parameter, which instructs sdStor what order to scan the library in, and what range of values to process. The order, start, and end values are separated by semicolons. Valid order values are: `id`, `title`, `rundate`, `who`, `appdate`, `appwho`, and `keyword`. A special order of “keywords” provides a list of unique keywords in the library. `Rundate` and `appdate` values must be entered in the format `YYYYMMDD`, such as `20030115` for January 15, 2003. In addition, the `rundate` value can also use `HHMM` after `YYYYMMDD`, such as `200301151500` for 3:00 PM on January 15, 2003.

Here are some examples:

| | |
|--|---|
| <code>-scan "id"</code> | Scans all records in the library in ID order. |
| <code>-scan "title;Aging;Aging"</code> | Scans all records in title order, starting and ending with those report titles starting with “Aging”. |
| <code>-scan "rundate;20030115;20030122"</code> | Scans records in run date sequence in the date range January 15 through January 22, 2003. |
| <code>-scan "keyword;invoice 6000;invoice 6500"</code> | Scans keywords in the range specified. |

Fields Displayed

By default, scans will display the `id`, `title`, `rundate`, `runtime`, and `pagecount` for each report, separated by a semicolon. If a keyword scan is conducted, then the keyword is added to this default display. If desired, the fields and/or separator can be changed.

If the scan order is “keywords”, then the only field displayed is a keyword. This scan order is designed specifically to list unique keywords in the library.

To change the fields displayed, use the `-fields` option. To change the field delimiter for output, use the `-sep` option.

| | |
|-------------------------------------|---|
| -fields "id;title;rundate" | Specifies three fields for output |
| -fields "id;title;rundate" -sep tab | Specifies the same three fields, delimited by tabs rather than semicolons |

The valid fields for display are id, title, rundate, runtime, who, appdate, appwho, memo, checksum, maxcol, maxrow, pagecount, and keyword, if the scan order is keyword.

Separators can be any character, or one of the special characters tab, lf, cr, or crlf.

Filtering Lines

In addition to the range specified with the -scan option, you can filter records so that only those matching the filter are displayed by the scanning process. For instance, in addition to scanning for a date range, you could filter for a specific customer name or a pattern. Filters are specified with the -filter option. A filter can be literal text or a regular expression prefixed with a tilde character. They are not case sensitive. Filters are applied to the fields being printed, so their effect can vary based on the -fields option.

| | |
|------------------------------------|---|
| -filter invoice | Filters for lines containing the word "invoice". |
| -filter "invoice 06500" | Filters for lines containing the phrase "Invoice 06500". Note the use of quotes due to the spaces in the phrase. |
| -filter "~invoice [0-9]+" | Filters for lines containing the regular expression of the word "invoice" followed by a space and one or more digits. |
| -filter "~(invoice) (credit memo)" | Filters for Invoice or Credit Memo. |

Pagination

On Unix, you can pipe the scan output through a more or pg filter. Optionally, you can specify a -page option to have sdStor control pagination for you. For example, you can pause the output every 24 lines by specifying -page 24.

Retrieving Reports

Once a report ID is known, you can retrieve that report or a selection of pages from it, simply by specifying the -id option on the command line. Specify the library to retrieve from using the -lib option, and your login, if necessary, using the -login and -password options. You must have read rights to the library in order to retrieve a report.

When a report is retrieved, it is printed to the file or device specified by the -o option, or to the standard console output. This output can be piped to a spooler on Unix. The

output can be formatted as plain text, pdf, or laser data. To specify pdf, use the `-pdf` option. Generally, this should coincide with the use of a `-o` option to a `.pdf` file name. To specify laser data, use the `-laser` option. When either pdf or laser formatting is specified, `sdStor` will create a document scaled to fit the columns and rows for a given report on a page. You can specify landscape orientation and paper size using the `-landscape` and `-paper` options, and the output format will honor the specifications. Plain text output does no such formatting, but is suitable for passing to other applications for processing.

If you specify laser formatting, be sure to set raw or pass-thru mode in any spooler options. For example, use `-oraw` on SCO Unix, `-b` on Linux, or `-o-dp` on AIX.

If you want to retrieve just specific pages of the archive, use the `-pages` option. This option takes a comma-delimited list of pages or page ranges. An artificially high page number will retrieve the last page.

If you have UnForm and wish to process the retrieved result through UnForm for enhanced output, you can specify the `-uniform` option. The UnForm command and rule file to use are configured in the `sdstor.ini` file. UnForm will be executed with the pdf driver if the `-pdf` `sdStor` option is used, or with the laser driver otherwise.

Report pages are normally delimited with formfeed characters. If you need the report produced with even numbers of lines per page and no formfeeds, specify the `-noff` option. This can be useful if a report is originally produced without formfeeds and is stored in the archive library with the wrong number of lines per page. You can retrieve the report with a `-noff` option and then re-archive it with the proper `-page` option or rule file page command.

Examples of report retrieval:

This will retrieve archive 100 from library ar, printing to the file out.txt:
`sdstor10 -lib ar -id 100 -o out.txt`

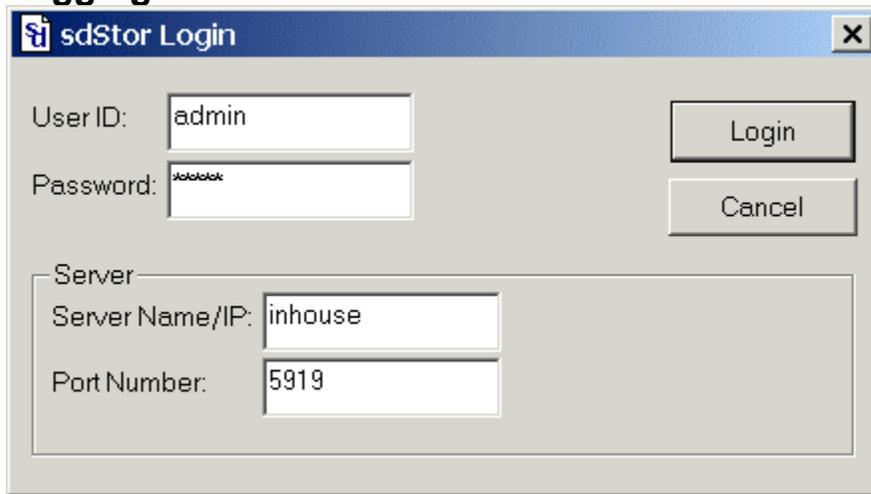
This will retrieve the same archive in pdf format:
`sdstor10 -lib ar -id 100 -pdf -o out.pdf`

This will retrieve the same archive and print it in laser format in landscape orientation:
`sdstor10 -lib ar -id 100 -laser -landscape | lp -dhp4000 -oraw`

sdStor Viewer

Once you have the View Server running, you can install the sdStor Viewer software on any Windows workstation on your network.

Logging In

A screenshot of the 'sdStor Login' dialog box. The window title is 'sdStor Login'. It contains several input fields and buttons. The 'User ID' field contains 'admin'. The 'Password' field contains a masked password '*****'. To the right of these fields are 'Login' and 'Cancel' buttons. Below these is a 'Server' section with a 'Server Name/IP' field containing 'inhouse' and a 'Port Number' field containing '5919'.

When launched, the sdStor Viewer will present a login form. Enter your user ID and password. When you press the Login button, the Viewer will connect to the server specified and validate your user ID and password. The server and port are stored for future logins.

User logins and passwords are defined by an administrator. Each login is also granted read, write, and/or delete access to specific archive libraries on the server. The options available to you as you search and navigate libraries depend upon the level of access granted to specific libraries.

Once logged in, you can fill in fields on the Search form and review results in the Scan window.

Search Form

The top section of the search form provides library selection, scan order, and range options. Select a field for the scan order, and enter a value range, if possible, for that scan order.

Performance Note: Libraries can potentially contain hundreds of thousands, or even millions, of records. It takes time to fully scan a large library. If you can limit scanning to a range of dates, titles, users, or keywords, then the scanning process can scan fewer records. For example, if you know the reports you wish to view were printed during a certain date range, then select a Run Date order and enter Start and End values that match that range.

The search criteria section allows you to specify words or phrases that must be present for the archive to be presented in the scan results. You can also specify a report content search. This provides greater flexibility to find specific reports, but does require that the report pages be uncompressed and searched, a time-consuming process.

The search criteria for the title, run user, report user, memo, and report content fields can be entered in three ways:

| | |
|---------------|---|
| Plain text | Searches for the case-insensitive occurrence of the specified text, such as “Invoice 006001”. |
| Text wildcard | Text that contains * or ? characters, such as “J*Smith”, is |

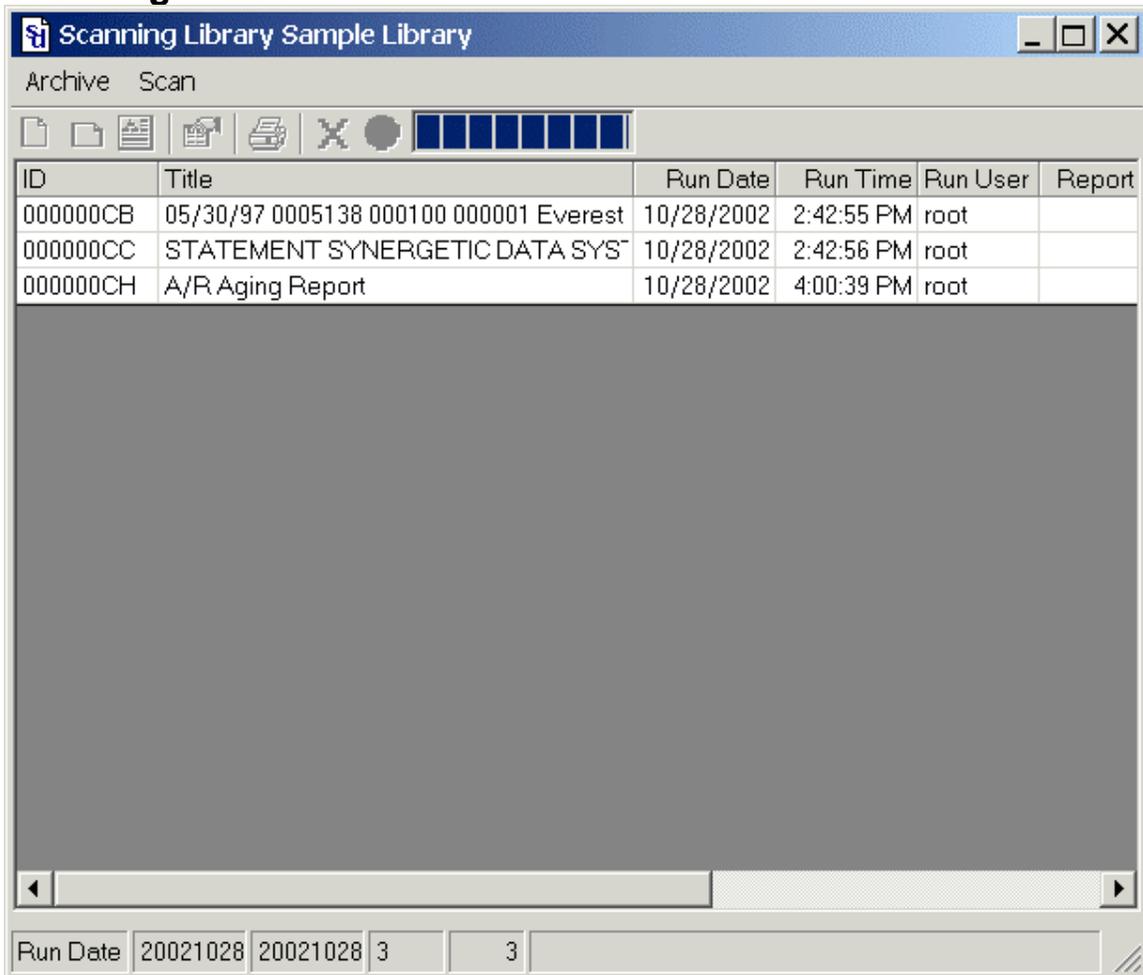
| | |
|--------------------|--|
| | treated like a wildcard, with * representing any set of zero or more characters, and ? representing any single character. If you want to search for literal * or ? characters, use * or \?, respectively. |
| Regular Expression | A regular expression preceded with a ~ character, such as “~\.99”, which treats everything after the ~ as a regular expression pattern. |

The search criteria for the two date fields can be entered in standard date format for your locale. For example, in the United States, you could enter the date as m/d/y, such as 3/25/2003. The Run Date field can be further qualified with time, such as “3/25/2003 3:00 PM”. You can enter time to a specific minute, or to a specific hour. If the time specified is a whole hour, such as 3:00 PM, then reports for the whole hour will be selected (3:00 through 3:59).

If more than one field criteria is entered, then all criteria specified must match for a given report to appear in the scan window.

Press the Search button to begin the scanning process.

Scanning Window



The scan window produces a list of report archives that match the search form criteria specified. A progress meter shows the status, and you can stop and continue the scan at any time. If the report you need has been displayed, then you can reduce the load on the server and reduce screen flicker by stopping the scan.

You can sort the columns by clicking the column header, and you can select reports and perform actions using the toolbar, archive menu, or right-click. You can select multiple reports by Control-clicking them individually, or by clicking the start of a range and Shift-clicking the end of the range. Bulk transfers and deletions are available when multiple reports are selected.

The actions available are:

| | |
|----------------|--|
| View | Standard portrait view of the report. Double-clicking the report will invoke this option. |
| View Landscape | Landscape view of the report. |
| View UnForm | Enhanced view as processed through UnForm's pdf driver, if UnForm is configured for use by sdStor. |

| | |
|--------------|--|
| Properties | If you have write rights to the library, you can edit the report properties, such as the title, keywords, and memo. |
| Transfer | If you have delete rights to this library, you can transfer this report to another library to which you have write rights. This option is also available if you have selected multiple reports. |
| Delete | If you have delete rights to this library, and if the View Server is configured to allow remote deletions, you can delete this report. It will go into the trash library, unless you are scanning the trash library, in which case it is permanently deleted. This option is also available if you have selected multiple reports. |
| Server Print | If server printing is configured in the View Server, then you can print this report without using the local viewer window. |

Report View Windows

05/30/97
03:40 PM

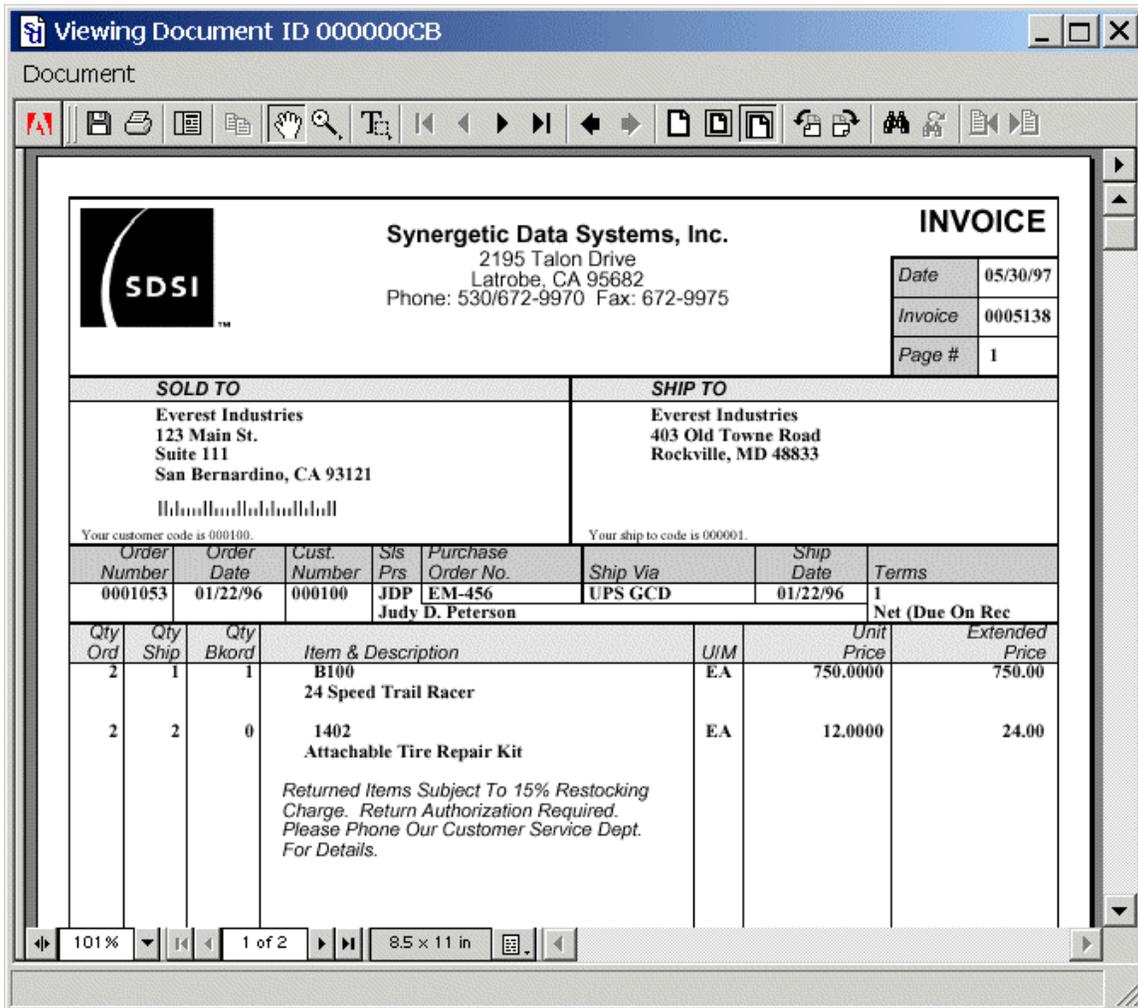
ADD-CB Manufacturing Company
Detail Aging Report As of 04/15/96
Based On Invoice Date
Alpha Sort From First To Last

Page 1

| Customer No. & Name | Invoice | Due | PO Number | Order Number | Terms | Type | Contact | Future | Current | 30-Days | 60-Days | 90-Days | 120-Days | Balance |
|-----------------------------------|----------------|---------------------|-----------|--------------|-------|------|---------|---------|---------|---------|---------|---------|----------|----------|
| 000500 Ron Anderson And Company | (619-552-6589) | Albert Posner | | | | | | | | | | | | |
| 0005080 | 02/15/96 | None | None | 5 | S | | | | | | 1309.27 | | | 1309.27 |
| 0005093 | 03/02/96 | None | None | 5 | S | | | 1470.73 | | | | | | 1470.73 |
| 0005105 | 03/08/96 | None | None | 5 | S | | | 2086.38 | | | | | | 2086.38 |
| 0005118 | 03/18/96 | None | None | 5 | S | | | 4034.16 | | | | | | 4034.16 |
| 0005001 | 12/27/95 | None | None | 5 | S | | | | | | | 28.67 | | 28.67 |
| 0005011 | 02/28/96 | None | None | 1 | F | | | | 5.00 | | | | | 5.00 |
| 0005022 | 03/30/96 | None | None | 3 | S | | | 1200.00 | | | | | | 1200.00 |
| 0A51222 | 12/22/95 | None | None | | | | | | | | | 28.67- | | 28.67- |
| | | | | | | | | 0.00 | 5234.16 | 3562.11 | 1309.27 | 0.00 | 0.00 | 10105.54 |
| 001001 Baker And Harrison | (206-955-3443) | Arnold Baker | | | | | | | | | | | | |
| 0005057 | 01/31/96 | None | None | CD | S | | | | | | 36.80 | | | 36.80 |
| 0005075 | 02/05/96 | None | None | CD | S | | | | | | 158.75 | | | 158.75 |
| 0005087 | 02/23/96 | None | None | CD | S | | | 629.00 | | | | | | 629.00 |
| 0005099 | 03/02/96 | None | None | CD | S | | | 1863.40 | | | | | | 1863.40 |
| 0005111 | 03/08/96 | None | None | CD | S | | | 592.91 | | | | | | 592.91 |
| 0005124 | 03/18/96 | None | None | CD | S | | | 1287.00 | | | | | | 1287.00 |
| | | | | | | | | 0.00 | 1287.00 | 3085.31 | 195.55 | 0.00 | 0.00 | 4567.86 |
| 000700 Douglas Erickson & Company | (619-452-2589) | Evelyn Stevens | | | | | | | | | | | | |
| 0005071 | 02/09/96 | None | None | 1 | S | | | | | | 273.75 | | | 273.75 |
| 0005095 | 03/02/96 | None | None | 1 | S | | | 1718.85 | | | | | | 1718.85 |
| 0005108 | 03/08/96 | None | None | 1 | S | | | 2152.87 | | | | | | 2152.87 |
| 0005120 | 03/18/96 | None | None | 1 | S | | | 1846.33 | | | | | | 1846.33 |
| 0005013 | 02/28/96 | None | None | 1 | F | | | 85.48 | | | | | | 85.48 |
| | | | | | | | | 0.00 | 1846.33 | 3987.20 | 273.75 | 0.00 | 0.00 | 6077.28 |
| 000100 Everest Industries | (714-832-8113) | Mr. James Rockville | | | | | | | | | | | | |
| 0005053 | 01/31/96 | None | 0001055 | 1 | S | | | | | | 1826.77 | | | 1826.77 |
| 0005058 | 02/02/96 | None | 0001060 | 1 | S | | | | | | 3272.00 | | | 3272.00 |
| 0005082 | 03/02/96 | None | 0001091 | 1 | S | | | | | 3257.23 | | | | 3257.23 |
| 0005101 | 03/08/96 | None | 0001104 | 1 | S | | | | | 3237.10 | | | | 3237.10 |
| 0005114 | 03/18/96 | None | 0001117 | 1 | S | | | | | 3563.98 | | | | 3563.98 |
| 0005127 | 03/30/96 | None | 0001130 | 1 | S | | | 3563.98 | | | | | | 3563.98 |
| 0005132 | 04/02/96 | None | 0001145 | 1 | S | | | 298.10 | | | | | | 298.10 |
| 0005133 | 04/02/96 | None | 0001146 | 1 | S | | | 298.10 | | | | | | 298.10 |
| 0005134 | 04/02/96 | None | 0001147 | 1 | S | | | 298.10 | | | | | | 298.10 |
| 0005135 | 04/05/96 | None | 0001148 | 1 | S | | | 298.10 | | | | | | 298.10 |
| 0005136 | 04/06/96 | None | 0001149 | 1 | S | | | 298.10 | | | | | | 298.10 |
| 0005017 | 03/04/96 | None | None | 3 | S | | | | 1200.00 | | | | | 1200.00 |
| | | | | | | | | 4990.50 | 5231.48 | 7694.33 | 5098.77 | 0.00 | 0.00 | 23015.08 |
| HEC001 Hillbore Bicycle Center | (503-555-1212) | J Bruce Thorsson | | | | | | | | | | | | |
| 0005113 | 03/15/96 | None | 0001116 | CD | S | | | 450.00 | | | | | | 450.00 |
| 0005126 | 03/18/96 | None | 0001129 | CD | S | | | 4502.23 | | | | | | 4502.23 |
| 0005128 | 03/30/96 | None | 0001131 | CD | S | | | 4045.00 | | | | | | 4045.00 |

78% 1 of 3 11 x 8.5 in

The standard view is a simple pdf document generated by the server. The viewer uses the Adobe Acrobat plug-in and an instance of Internet Explorer for viewing.

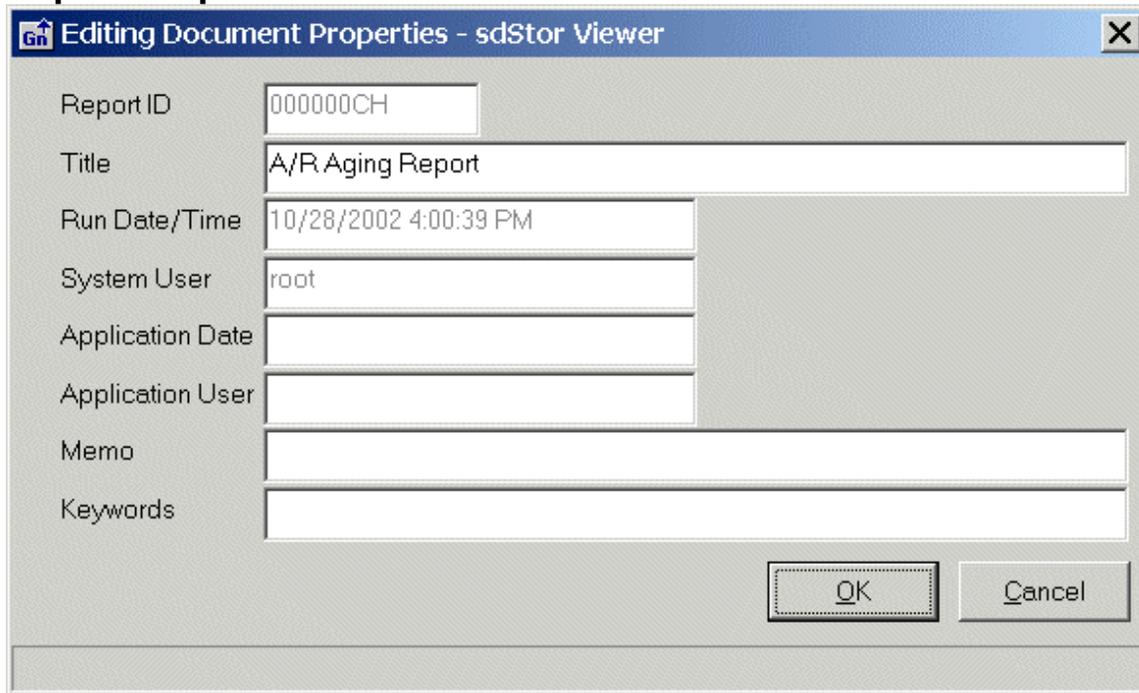


If sdStor is configured to work with UnForm, the View Server can generate enhanced pdf files for the viewer. This is particularly powerful when retrieving forms, where the plain text data doesn't represent the full report.

From the viewer, you can perform the following actions:

| | |
|--------------|---|
| Print | Print the document to a local Windows printer |
| Save As | Save the report pdf to a local file |
| Email | Email the report pdf as an attachment |
| Properties | Edit report properties, if you have write rights to the library. |
| Transfer | Transfer the archive to another library, assuming you have proper rights. |
| Delete | Delete the archive, assuming you have proper rights. |
| Select Pages | Specify a limited set of pages to view, and refresh the display. You can also use this option to print a crosshair pattern on the selected pages. |

Report Properties



The screenshot shows a dialog box titled "Editing Document Properties - sdStor Viewer". It contains the following fields and values:

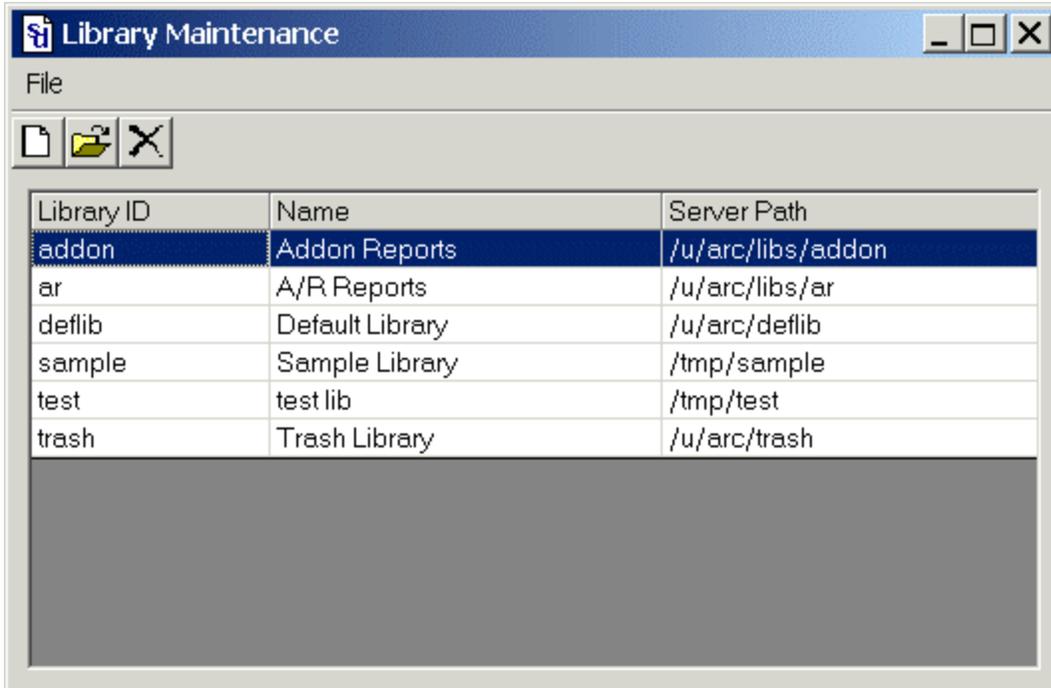
| Field | Value |
|------------------|-----------------------|
| Report ID | 000000CH |
| Title | A/R Aging Report |
| Run Date/Time | 10/28/2002 4:00:39 PM |
| System User | root |
| Application Date | |
| Application User | |
| Memo | |
| Keywords | |

Buttons: OK, Cancel

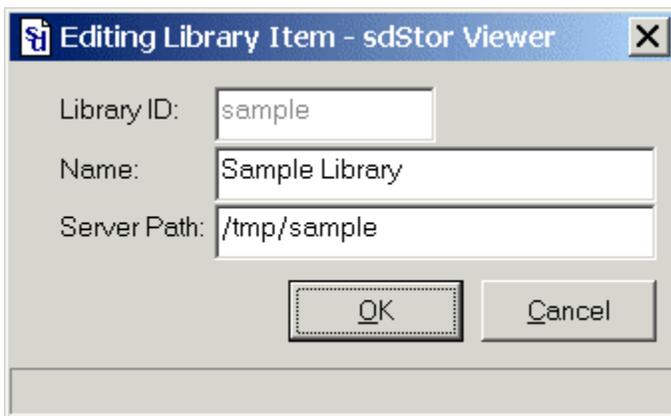
If your login has write access to the library where the report is stored, you can edit the report properties. The title, application date, application user, memo, and keywords fields are all editable. Keywords are short words or phrases that can be used to quickly locate this report in a keyword-ordered search. There can be any number of keywords for any report, delimited by a semicolon (;). If you need to include a semicolon in a keyword, then use the sequence \; instead.

Administration Tools

If you have logged in as an administrative user, the Search window will provide access to Library, User, and User-Library Access maintenance functions. Each of these three functions works in a similar manner. A list of current table entries is presented, and you can select one to edit or delete, or can create a new entry.



The above image shows a list of libraries. Double-clicking an entry will launch an edit window, like the one below.



Note: when maintaining libraries, the path specified must be valid for the server, not the client. Be sure to avoid characters that would be invalid in file names on the server.

Automating Report Properties with Rule Files

Rule files are text files that contain commands to automatically set archive data and library specifications for reports that meet specific criteria. There can be any number of these specifications, called *rule sets*, in a rule file. A header line composed of a unique name enclosed in square brackets indicates a new rule set. For example, an invoice rule set would begin with the line "[Invoice]", followed by several lines indicating how to archive invoice data. Without a rule set to work with, or specific command line parameters, sdStor will assume a simple default title based upon the initial content of the report, but that is all. A rule set enables the precise definition of title, keywords, memo, and other user-defined fields. sdStor determines which rule set to work with based on either a command line option (-r) or **detect** commands contained in the rule set.

Following the [*form-name*] line are commands and (usually) a list of parameters separated by commas. The available commands are described on the following pages.

Unless otherwise noted, all column and row specifications are 1-based (i.e. the first column is 1, rather than 0).

Commands that have parameters accept either a space or an equal sign between the keyword and the first parameter; **page 66** and **page=66** are equivalent.

If a command and its parameters require a large amount of text, it is possible to split a command across multiple lines by adding a backslash character at the end of a line to indicate the command continues on the next line. You can have as many continuation lines as necessary. sdStor removes leading spaces and tabs from continuation lines, so you can use indentation to improve readability, as long as you remember to place any required spaces before the backslash on the initial line. For example:

```
Title {“This is a very long title consisting of some text plus an expression ” \  
      +trim(get(1,2,80))}
```

Note that any errors in expressions or code blocks are reported only to the error log file specified with the -e command line option.

APPDATE

Syntax

appdate “*date*” | {*expression*}

Description

The application report date is a user-defined field that can be used to specify a report date, which may be different from the run date. Application report dates may be based on fiscal period or month end dates, which may not match the system date at the time the report is executed. sdStor provides the appdate field for these dates.

The *date* or the result of the *expression* must be in delimited or fixed date format. Fixed date format is a string of 4, 6, or 8 digits. Delimited data format is a three-part string of digits separated by non-digit characters. The order in which the digits are interpreted is determined by the datefmt value, which can be specified in the rule set, the command line, or in the user’s sdStor configuration file.

The date format defaults to “mdy”, but can be set to “dmy” or “ymd”. If the year is not specified, it defaults to the current system year. If the year is specified as two digits, the century defaults to the current system date’s century.

If an *expression* is provided, it must evaluate to a valid date string.

Example

Appdate {get(1,1,8)}

APPWHO

Syntax

appwho “*user*” | {*expression*}

Description

The application user is a user-defined field that can be used to specify an application user ID, which may be different from the system login user ID. Accounting applications often have their own login process, and report user codes on reports. These codes can be specified in addition to the system login code automatically captured.

If an *expression* is provided, it must evaluate to a string.

Example

Appwho {trim(get(73,1,8))}

CODE BLOCKS – PAGE{}, JOB{}

Syntax

```
job | page {  
  code block  
}
```

Note: the opening brace “{” needs to be on the same line as the keyword. The closing brace may follow the last statement, or be on the line below the last statement.

Description

These keywords are used to add Business Basic processing code to the form or report. They represent two different subroutines that sdStor executes at specific points during processing. The *code block* can be an arbitrary number of Business Basic statements; the total number of statements in all code blocks can be about 6,000 (or less, depending on program size limits imposed by the run-time environment).

Note that errors in code blocks are only reported to the error log file specified on the command line with the `-e` option. When developing and testing rule files, using the error log file option is therefore recommended.

- **job** executes after the last page has been read from the input data stream. Use this for any final cleanup or assignment of variables. At the end of the job, after this code block is executed, the report is archived.
- **page** executes after each page is read. Use this to gather data associated with any page.

Any valid Business Basic programming code can be entered, including I/O logic, loops, variable assignments, and more. sdStor will add extensive error handling code within your code, and report syntax errors to the error log file.

You may use the following variables and functions in your *code block*:

- **text\$[all]** is a one-dimensional array of the text for the page. For example, text\$[2] is the second line of text on the page.
- **skip** may be set to a non-zero value in the job code block in order to skip archiving of this report, or in a page code block to skip archiving of that page.
- **pagenum** contains the current printed page number during page code blocks.
- The following variables may be set to perform the same functions as the commands of the same name. When set in code blocks, these variable values take precedence over rule set command values.

- AppDate\$
- AppWho\$
- Datefmt\$
- Docid\$
- Keywords\$
- Kwsep\$
- Library\$
- Memo\$
- Title\$
- **mid(*arg1*,\$*arg2*,\$*arg3*)** is a function that safely returns a substring without generating an error 47 if the value in *arg1*\$ isn't long enough to accommodate position *arg2* and length *arg3*.
- **get(*col*,\$*row*,\$*cols*)** is a function that safely returns text from the text\$[all] array, without substring or array out-of-bounds errors.
- **mget(*col*,\$*row*,\$*cols*,\$*rows*,\$*dml*,\$)** returns multiple lines of text into a single string, delimited by the *dml*\$ value, or a semicolon if *dml*\$ is null. This function is useful for generating keywords from a region on a page.
- **err=next** may be used for any *err=label* option in any function or statement, in order to force UnForm's error trapping to ignore an error. You may, of course, name your own *err=label* if desired.
- **trim(*expression*)** trims spaces from the left and right side of a text expression.
- **upper(*expression*)** converts text to UPPERCASE.
- **lower(*expression*)** converts text to lowercase.
- **proper(*expression*)** converts text to Proper Case.
- **cnum(*expression*)** returns a number from a text string, after stripping formatting characters such as commas and dollar signs. Parentheses and minus signs indicate negative numbers.

When using variables and line labels, you should avoid using any values that begin with "AR". SdStor reserves all such variables and labels for its use. You may use a backslash (\) at the end of a line to continue the statement on the next line. Lines prefixed with "#" are not added to the code.

Note that the merge command, while not executable code, is honored within a code block. The merged data must be valid code block syntax.

For more details about programming code blocks, see the Programming Fundamentals chapter.

Example

This example shows how to assign a title and custom keywords to a job:

```
Page{  
  
If pagenum=1 then title$=trim(get(20,2,60))  
Keywords$=keywords$+"customer "+get(10,10,6)+";"  
  
# Scan looking for high value lines to record "high dollar" keyword  
for row=1 to 66  
  if cnum(get(80,row,10))>100000 then \  
    if pos("high dollar"=keywords$)=0 then \  
      keywords$=keywords$+"high dollar;"  
next row  
  
}
```

CONST

Syntax

`const ID=value`

Description

The **const** keyword provides the capability to use a named value as a parameter to other keywords. A given constant ID can be reused, and references to it in subsequent rule set lines will reflect the new value. In addition, a constant defined before the first rule set in the rule file will apply to any rule sets in the file, unless the same ID is reused in any particular rule set.

NOTE: Case does make a difference. “COLPOS” and “colpos” are different constants. Take care not to use constant names that may inadvertently cause replacements elsewhere than intended.

Constant names are limited to 25 characters, and constant values are limited to 75 characters. If you use a quoted value, the outer quotes are removed before the value is substituted into the rule file commands.

Example

Const DTF=mdy would set a constant value to “mdy”, which could then be used throughout the rule set in `datefmt` commands, like **datefmt “DTF”**.

DATEFMT

Syntax

datefmt “*date*” | {*expression*}

Description

Application dates can be specified using the appdate command. The structure of application dates can vary depending on the date format setting. Date formats determine the order in which data elements are parsed: mdy, dmy, or ymd.

The date format can be set with this command in the rule set, or with the command line -datefmt option, or with the datefmt= line in the users sdStor profile.

If an *expression* is provided, it must evaluate to a string of “mdy”, “dmy”, or “ymd”.

Example

Appdate “dmy”

DETECT

Syntax

```
detect column,row,"text|!=text|~regex|!~regex"
```

Description

This option is used to identify a job from the first page of data read by sdStor. If the **-r** option is used on the command line, then **detect** keywords are ignored. Otherwise, each rule set's detects are analyzed until a match is found. If more than one **detect** keyword is specified for a rule set, then the form must match all of them. Detection occurs only at the start of the job, using the first page of data read from the input stream.

If *column* and *row* are 0, then the whole page is scanned for the occurrence of the text. If *column* is 0, then the whole line is scanned.

Column and *row* can contain ranges in the format *from-through*, such as '20-25' for the columns (or rows) 20 through 25.

The format of the quoted match text determines how the detection scan is handled. If plain text is specified, then a literal match for *text* is performed. If the text begins with the prefix character ~, then a regular expression search for *regex* is performed. If *text* begins with the string "!=", or *regex* begins with "!~", the detect scans for NON-matches in the region specified. For example, 'detect 1,1,"!=INVOICE"' would detect any document except one that contains the text "INVOICE" at column 1, row 1.

Examples

detect 0,2,"INVOICE" would search for INVOICE anywhere on line 2.

detect 10-12,4,"~../.." would match a date format at column 10, 11, or 12, on row 4.

detect 65-66,6-8,"!~../.." would match a date format NOT occurring at column 65 or 66, on rows 6 through 8.

DOCID

Syntax

docid {*expression*}

Description

This command sets the document ID code to the result of the *expression*. The expression is evaluated as each page is read from the input stream. The document ID for each page is compared to the prior document ID generated for the job, and whenever the ID changes, a new archive is generated. This is useful for breaking a batch job, which may contain many documents, into multiple archives. An example of a document ID might be an invoice number during an invoice batch print, or a customer number during a statement batch print. Note that document IDs are not the same as page numbers. A customer's statement may run for several pages, and all those pages should generally be in the same archive record.

Whenever the document ID changes, all properties variables, such as title\$ and keywords\$, are set to null. Only a page{ } code block should be used to manage property variables when a document ID is used.

Beware of print jobs that have an extra line on pages after the first. This is a somewhat common issue with programs that print plain-paper versions of forms, printing a formfeed-linefeed combination is printed between pages, but not on the first page.

Example

This example will derive the document ID from column 60, row 6, for 8 characters:

```
Docid {get(60,6,8)}
```

This example will derive the document ID from line 6 on page 1, and line 7 on pages 2 and above:

```
Docid {get(60,6+(pagenum>1),8)}
```

KEYWORDS

Syntax

keywords “*words*” | {*expression*}

Description

Keywords are used to supply any number of indexes for report scanning and retrieval. Each keyword becomes a search key for the report. For any given report, there may be no keywords or hundreds, depending on the needs of the application and how easy retrieval needs to be. Designing keyword strategies takes some thought. Often they will be document identifiers, like “invoice xxxx” or “po yyyy”. If there are batch runs that contain many documents, the keywords field can consist of all the document IDs contained in the batch. Keywords may also contain customer or vendor names, or internal codes.

Each keyword is limited to 30 characters. Each keyword in the keywords string is delimited by a semicolon or by the separator specified in the KWSEP command. Keywords are not case sensitive.

If an *expression* is provided, it must evaluate to a string of delimited keywords.

Example

Keywords {“invoice “+get(70,6,6)+”,”+trim(get(10,11,30))}

KWSEP

Syntax

kwsep “*sep*” | {*expression*}

Description

Specifies the keyword separator used to parse the keywords field into individual keywords for indexing. The default separator is a semicolon.

If an *expression* is provided, it must evaluate to a string.

Example

Kwsep “##”

LIBRARY

Syntax

library | lib “*library ID*” | {*expression*}

Description

The library specifies where the report archive is stored. If a library is specified, then the report is stored there rather than in the default library, or one named on the command line. The library ID specified must exist in the libraries table maintained by administrative users.

If an *expression* is provided, it must evaluate to a valid library ID.

Examples

library “payroll”

library {“ar”+date(0:”%Y1”)} (dynamically sets library to ar2002, ar2003, etc.)

MEMO

Syntax

memo “*text*” | {*expression*}

Description

The memo field provides a place to store user-defined comments with the archive. The memo field can be up to 100 characters long.

If an *expression* is provided, it must evaluate to string.

Example

MERGE

Syntax

merge “*ruleset*” [, “*rulefile*”]

Description

This command will insert the contents of the *ruleset* into the currently parsed rule set. If the *rulefile* parameter isn't supplied, the current rule file is used. Otherwise, *rulefile* is opened in the sdStor directory or by full path, if specified, and is scanned for *ruleset*. This command can be used to incorporate common elements into many rule set formats.

Note that if no *rulefile* is specified, then the rule file specified for the job is used for the merge, even if the merge is nested within another merge that specifies another rule file.

Unlike other sdStor commands, merge works within a code block, such as precopy or prepage.

PAGE

Syntax

page *rows*

Description

This command specifies an input page length of no more than *rows* lines. If a form-feed character is encountered first, then the page is considered complete also. This keyword is useful if the application creates a form with line-feeds rather than form-feeds.

If no formfeed page delimiters are found in the input data stream, and no page command is specified, then sdStor will use a default page size of 66 lines.

Example

page 42 will consider each 42 lines to be a full page.

TITLE

Syntax

title “*title*” | {*expression*}

Description

The title of a report archive is an important way of identifying the report. Scans can be ordered by title for quick searching. The title can be up to 60 characters long.

If an *expression* is provided, it must evaluate to a string.

Example

Title “A/R Aging Report”

Title {trim(get(2,20,60))+} “+trim(get(1,20,60))}

Rule File Examples

Included with the sdStor installation is a sample rule file, `sample.rul`, and several report samples that work with the rule sets in `sample.rul`. To execute these samples, just execute a command line like this:

```
sdstor10 -i sample1.txt -f sample.rul -o test.out
```

Note that in this format, the sample will be archived to the Default library. If you want to archive them to a test library, you could set up such a library and add a `-lib testlib` option to the command line.

Each of the four rule sets has a comment that indicates which sample text file it is designed for. For example, the Invoice rule set uses `sample1.txt`, and the Statement rule set uses `sample2.txt`. The following pages document the various sample rule sets in `sample.rul`.

Invoice Rule Set – sample1.txt

All rule sets are prefixed by a line with a name in square brackets, such as [Invoice]. The name can be virtually any text, but generally should be indicative of the type of job it handles.

```
[Invoice]
# uses sample1.txt
```

Generally, detect logic will be used to automatically select a rule set when the job is processed. If no detect logic is specified, it is still possible to select a rule set from the command line. For example, adding “-r Invoice” to the sdstor10 command would select this rule set, and detect commands would be ignored. If a job doesn’t match the detect specifications, then the next rule set is tested, and so on. If no rule set is selected, the job is processed using default properties.

In this example, two patterns are tested: one at column 65, row 5, with a pattern of a date, two spaces, and additional characters, and the second at column 10, row 21, with a date, one space, and some characters.

```
detect 61,5,"~../... .."          # invoice date and #
detect 10,21,"~../... .."         # ord date and cust code
```

The next line sets the title for the job to the literal text “Invoice Run”.

```
title "Invoice Run"
```

The page code block executes several Business Basic lines as each page is read from the input stream. In this case, the code block is creating keywords for the customer number and item codes. The item codes are found in a column, in rows 25 through 56, but only when there is a quantity associated with the row.

```
page{
# keywords for customer codes and part numbers
keywords$=keywords$+"customer "+get(9,11,6)+";"
for row=25 to 56
  if cnum(get(1,row,3))>0 then \
    keywords$=keywords$+"item "+trim(get(22,row,20))+";"
next row
}
```

Statement Rule Set – sample2.txt

```
[Statement]
# uses sample2.txt
```

This rule set uses two detect statements. First, the literal STATEMENT must be at column 34, row 2, and second, a pattern of “PAGE:”, a space, and a digit, must be anywhere on line 5.

```
# detect two conditions, a literal title and a pattern
detect 34,2,"STATEMENT"
detect 0,5,"~PAGE: [0-9]"
```

If a report does not contain formfeeds between pages, sdStor should be told how many lines are printed per page. In this case each page is 66 lines.

```
# set number of lines per page, since this job has no formfeeds
page 66
```

The title command sets the title for the archive.

```
title "A/R Statements"
```

The page code block executes several Business Basic lines as each page is read from the input stream. In this case, the code block is creating keywords for company names in the job. To account for the possibility of semicolons in the names, a different keyword separator is used, in this case, a tab character (ASCII character 9). The customer names are found at column 12, row 11.

```
page{
# build keywords using customer names
tab$=chr(9)
kwsep$=tab$
compname$=trim(get(12,11,30))
if pos(compname$+tab$=keywords$)=0 then \
    keywords$=keywords$+compname$+tab$
}
```

AR Aging Rule Set – sample3.txt

```
[AR Aging]
# uses sample3.txt
```

This report has a single, simple detect statement, looking for the report title at column 50, row 2.

```
detect 50,2,"Detail Aging Report"
```

The title is derived from the report itself, by using an expression with the get() function.

```
title {trim(get(40,2,50))}
```

The job code block is executed once, after all pages have been read. The last page is still available in the array text\$[] and the get() function returns the last page's data. In this sample, the application date is printed in the upper-left corner of the page. The date, however, only contains a 2-digit year, so some additional logic is used to add a century based. Years 0 through 50 are assumed to be in the 2000 century, while years 51 through 99 are assumed to be in the 1900 century. In addition, the memo field is field is set to the combination of two additional heading lines.

```
job{
# set some variables - in particular handle Y2K date issue
appdate$=get(1,1,8)
if cnum(mid(appdate$,7,2))>50 then \
    century$="19" else \
    century$="20"

appdate$=mid(appdate$,1,6)+century$+mid(appdate$,7,2)
datefmt$="mdy"
memo$=trim(get(40,3,50))+ " "+trim(get(40,4,50))
}
```

Purchase Orders Ruleset – sample4.txt

```
[Purchase Orders]
# use sample4.txt
```

The detects for this job are two literal values, one at column 44, row 2, and the other at column 88, row 3.

```
detect 44,2,"PURCHASE ORDER"
detect 88,3,"NO."
```

This example uses the docid command to split the job into multiple archives. Each time the purchase order number changes, a new archive is generated. The purchase order is found at column 92, row 3, for 6 characters.

```
# break job into individual documents as PO number changes
docid {get(92,3,6)}
```

The page code block is used to set the title by defining the variable title\$. In addition, keywords for each inventory item are created in a loop, looking for item codes in column 23, rows 22 through 59. The keywords and title values, as well as the pagenum reserved value, are reset for each document archive generated.

```
page{
title$="Purchase Order "+get(92,3,6)

# set some keywords for vendor and item codes
if pagenum=1 then keywords$="Vendor "+trim(get(25,10,10))+";"
for row=22 to 59
  item$=trim(get(23,row,15))
  if item$>" " then keywords$=keywords$+"inv item "+item$+";"
next row
}
```

Command Line Administration

Administration tasks for sdStor include managing user logins, libraries, and user-library access rights. Some additional tasks are often limited to administrative users, such as delete access to the trash library, but may be configured for other users as well.

Administrative users are automatically granted complete access to all libraries for retrieval, transfer, and delete operations. Therefore, there is no need to grant an administrative user specific access rights to certain libraries. Care should be taken, of course, to maintain confidentiality of the login and password codes for administrative users.

That said, it should be noted that there must be at least one administrative user defined in the users table, and if there isn't one, such as at initial installation, then a default one is created with the login of "admin", password "admin". One of the first tasks for maintaining a secure archive system is to change this password or define a new administrative user and delete admin.

The following operations are restricted to administrative logins.

User Table

The user table consists of records with four fields: ID, Name, Password, and Admin. These can be edited easily using the sdStor Viewer, or can be maintained from the command line using these commands:

To generate a list of current users, use the `-userlist` option, optionally with a `-sep` option to override the default semicolon delimiter. For example, to produce a tab-delimited list of users:

```
sdstor10 -userlist -sep tab
```

To edit or add a user, use the `-useredit` command with the four required fields, ID, Name, Password, and Admin. Admin can be set to 0 for a regular user or 1 for an administrative user. Fields are delimited by semicolons. If the user ID exists, the record is updated. If not it is added. For example, this will add or update the login record for adams:

```
sdstor10 -useredit "adams;Adam Smith;buggles;0"
```

To delete a user, use the `-useredit` option with just the user ID, like this:

```
sdstor10 -useredit "adams"
```

Library Table

The library table consists of records with three fields: ID, Name, and Path. These can be edited easily using the sdStor Viewer, or can be maintained from the command line using these commands:

To generate a list of current libraries, use the `-liblist` option, optionally with a `-sep` option to override the default semicolon delimiter. For example, to produce a tab-delimited list of libraries:

```
sdstor10 -liblist -sep tab
```

To edit or add a library, use the `-libedit` command with the three required fields, ID, Name, and Path. The path points to a directory and base file name. The directory must exist, and the path must be a valid file name for the operating system. For example, the name may not contain shell-specific characters, such as `&` or `[`, on Unix. The first time a report is archived to that library, sdStor will create files using the directory and base name and extensions `.ar1`, `.ar2`, and `.ar3`.

The three fields are delimited by semicolons. If the library ID exists, the record is updated. If not it is added. For example, this will add or update the library record for `ar2003`:

```
sdstor10 -libedit "ar2003;A/R Reports for 2003;/u0/libs/ar2003"
```

To delete a library, use the `-libedit` option with just the ID, like this:

```
sdstor10 -libedit "ar2000"
```

When you delete a library, you are only deleting the library definition, not the archive files. To completely delete a library from the system, you would also need to delete the `.ar1`, `.ar2`, and `.ar3` files of the library path. You can also change the name of the library accessing that path by simply adding a new library ID using that same path.

User Library Access Table

This table consists of records with three fields: User ID, Library ID, and Access. These can be edited easily using the sdStor Viewer, or can be maintained from the command line using these commands:

To generate a list of current records, use the `-userliblist` option, optionally with a `-sep` option to override the default semicolon delimiter. For example, to produce a tab-delimited list of user library access records:

```
sdstor10 -userliblist -sep tab
```

To edit or add a record, use the `-userlibedit` command with the three required fields, User ID, Library ID, and Access. Access can be set to 1 for read access, 2 for read/write access, or 3 for read/write/delete access. Fields are delimited by semicolons. If a record exists for the User ID and Library ID, the record is updated. If not it is added. For example, this will add or update the record for user login "adams", library "ar2003", granting read/write access:

```
sdstor10 -userlibedit "adams;ar2003;2"
```

To delete a record, use the `-userlibedit` option with just the User ID and Library ID, like this:

```
sdstor10 -userlibedit "adams,ar2003"
```

Managing Library Data Files

All archives are stored in library data files on a hard disk. These files are located in the path specified with the library was first defined, and have suffixes `.ar1`, `.ar2`, and `.ar3`. The library definition record contains a path that is a pointer to the disk location of these files. When you manage the files, you must also manage the library definition, and visa versa.

Be sure that when you perform archive data file management, users are not capturing archives or retrieving reports to the libraries you are managing. Remember also that it is important to treat the report archive files as you would any other important data file, and back them up regularly.

If you need to move a library, you need to move all the data files together, ensure read/write permissions, and then change the library path to match the new location. For example, on Unix:

```
mv /usr/archives/ar_reports.ar[123] /u9/archives/  
chmod 666 /u9/archives/ar_reports.ar[123]  
sdstor10 -libedit "arreports;A/R Reports;/u9/archives/ar_reports"
```

If you wish to move an archive so it is no longer written to, but is available for report retrieval, then you can simply move the data files to another location and define a new library to point to this new path. The original library will be created with a new set of archive data files the first time it is used. The new library definition can be used for

querying for any users with read or greater rights. If you need to prevent users from being able to add new records, simply set the file permissions to read-only.

As archives are deleted from libraries, their space is available for other archives that are added later. However, the space is not returned to the operating system. If a library has grown very large and has many deleted records, it may be desirable to reclaim the space. To do so, you can use the `-rebuild` option, which creates a new version of the archive with only the records in use. It is a good idea to make a backup of the library data files before performing a rebuild. If there is a system failure during this process, the library might be unusable.

Security

Report security is a key element of report retrieval in sdStor. The security model is based upon a login and password, and a table of access modes to specific libraries. Logins codes, libraries, and login-library access modes are be maintained by an administrative user through either the command line or the Windows interface. Once established, sdStor and the Windows server honor access modes to limit read, write, and delete operations to users who have proper access rights.

Login security is required for any retrieval or administrative task. It is not required for creating an archive: any user who can print reports to a configured printer can generate the archive to any library specified in the command line or rule file. However, a security login is required to retrieve a report archive, edit its properties, or transfer or delete it.

Access levels of read, read/write, and read/write/delete are granted to a user login and library combination. In order to retrieve a report, library read access is required. In order to update report properties, write access is required. In order to transfer a report, delete access is required for the original library, and write access is required to the destination library. To delete a report from any library other than trash, the user needs transfer rights to delete from the original library and write to the trash library. To delete archives from the trash library, delete rights are required.

In the sdStor Viewer, an initial login is required to start a session. All library and menu option access is controlled using the login user ID.

When working from the command line, however, a login is validated for each command entered. In this case, the login is accomplished in one of these ways, listed in order of precedence:

- Using the `-login` and/or `-pswd` options on the command line. The `-login` option can supply both user ID and password in the format *userID/password*.
- Using `login=` and `pswd=` lines in the user's sdStor defaults file. This file is named `sdstor` by default, in the directory specified by the `$HOME` environment variable. It can also be specified as a full path in the `$SDSTORDEFS` environment variable. This file should be owned by the login user and given read/write owner permission and no non-owner permissions (`chmod 600`). This file can also specify the user's default library (`deflib=` line) and date format (`datefmt=mdy|dmy|ymd` line).
- If no login is found by the above two methods, then the `sdstor.ini` file defaults section is searched for `login=` and `pswd=` lines to use.

For information about setting up user logins, libraries, and user library access, see the Administration chapter. Be aware that initially upon installation, there will be a default administrator login of admin/admin, and any time sdStor detects there is no other administrator login, this default login will be created. For that reason, one of the first security tasks is to use the admin/admin login to create a new administrative user and delete the admin user, or else change the admin password.

Regular Expressions

Regular expressions are supported in library searches, as well as in detect statements in rule files. Regular expressions are similar to, but much more powerful than, MS-DOS or UNIX *wildcards*. Regular expressions are indicated in searching or detect statements by a ~ prefix. For example, **detect 0,1,"~(INVOICE)|(CREDIT MEMO)"** searches for the regular expression (INVOICE)|(CREDIT MEMO) on line 1.

A regular expression is used to match patterns in text. By using special characters, called *meta characters*, sdStor can be instructed to search for patterns, such as dates or codes, and use them in processing. Below is a description of the various meta characters and how to use them.

- The simplest regular expression contains no meta characters. It just matches itself. **John** will match any occurrence of the text “John”.
- Brackets can be used to match any of a group of values: **[Jj]ohn** will match both “John” and “john”.
- If a range of letters or numbers is valid in a position, then the range can be indicated in a similar manner: **[A-Za-z]ohn** will match any letter, upper or lower case, followed by the letters “ohn”.
- If single character positions are not enough, then groups of options can be used with parentheses and vertical bars, like this: **(John|Jack|Jill) Smith**, which matches any of the first names, along with “Smith”.
- If any character will do in a position, use a dot: **Jo.n** will match “Jo”, followed by any single character, followed by “n”.
- To repeat any pattern, including a dot, use an asterisk (*) for 0 or more repetitions, or + for 1 or more repetitions: **J.*n** will match a “J”, followed by 0 or more characters, followed by “n”. **Jo+n** would match a “J” followed by one or more “o”s, followed by “n”.
- You can include multiple meta characters and patterns in the expression. For example, to search for 3 digits followed by 2 letters: **[0-9][0-9][0-9][A-Z][A-Z]**.
- To disable the special meaning of any of the meta characters, prefix it with a backslash. For example, a phone number might include parentheses; to include them in the expression, they must be disabled: **\(...\)--...-....**
- The meta characters are: ., *, +, (,), |, [,], ^, and \$.

Command Line Tutorial

The following series of examples show how to use the command line interface to administer an sdStor installation, and add and retrieve reports. If your installation is on Windows, you can enter these commands from a MS-DOS Command Window in the sdStor directory. Wherever directory paths are used, such as the sample archive path, you can substitute local C: paths, using either forward or backward slashes.

Change admin password

Change the default administrator login password to “12345”, or whatever value you like.

Command: **sdstor10 –useredit “admin;Admin User;12345;1” –login admin/admin**

Each command requires a –login option, or you can define a sdStor defaults file (usually .sdstor in your home directory) with standard login and password information. See the Security chapter for more details.

Add a regular user

Add a non-administrative user who will be able to read, but not write or delete, the sample library.

Command: **sdstor10 –useredit “sampuser;Sample User;123;0” –login admin/12345**

Add sample library

Add a sample library to hold the report archives for the tutorial.

Command: **sdstor10 –libedit “sample;Sample Reports;/tmp/sample”**

This and further examples assume you have defined a sdStor defaults file and no longer need a –login option for each command line.

Add read rights for sample library to the regular user

Establish “read” rights (level 1) for the user “sampuser”, created above, to the library “sample”, created above.

Command: **sdstor10 –userlibedit “sampuser;sample;1”**

Capture reports to the sample library

sdStor can capture incoming text reports to the sample library, simply by using the appropriate options. It will send a copy of the report on as output. These examples send the output to a file call sdstor.out. On Unix, you can use pipes and redirection:

```
cat sample1.txt | sdstor10 -lib sample >sdstor.out
cat sample2.txt | sdstor10 -lib sample >sdstor.out
cat sample3.txt | sdstor10 -lib sample >sdstor.out
```

On Windows or Unix, you can use command line options:

```
sdstor10 -i sample1.txt -o sdstor.out -lib sample
sdstor10 -i sample2.txt -o sdstor.out -lib sample
sdstor10 -i sample3.txt -o sdstor.out -lib sample
```

You can repeat these commands as many times as you wish. Each time, the report will be archived in the sample library.

Scan the library

If you have defined a user .sdstor file that contains a valid login and password, or you have defined a default login and password in the sdstor.ini file, then you won't need to add a -login option to the scanning commands. If necessary, you can add '-login sampuser/123' to each command, and you will be logged in as the sample user defined above, which has been granted read rights to the sample library.

Command: **sdstor10 -lib sample -scan id -fields "id;title;rundate;runtime"**

The result of this command, with the three above captured reports stored, would look like this:

| |
|---|
| ID;TITLE;RUNDATE;RUNTIME 000000CB;05/30/97 0005138 000100 000001 Everest Industries Everest In;10/28/2002;14:42:55 000000CC;STATEMENT SYNERGETIC DATA SYSTEMS INC. 2195 TALON DRIVE PAGE;10/28/2002;14:42:56 000000CD;05/30/97 ADD+ON Manufacturing Company Page 03:40 PM Detail A;10/28/2002;14:42:56 |
|---|

The ID codes on your system will vary from these examples. In the absence of a rule set to pre-define the title format, the default titles you see are derived from the initial words found in the document. You can use several fields for scan ranges, including ID, title, run date, and more. To scan a range, specify the range field and values in the -scan option, as in this example:

Command: **sdstor10 -lib sample -scan "title;statement;statement" -fields "id;title;rundate"**

Note that only report(s) whose title starts and ends with “statement” are displayed. This is due to the arguments provided to the -scan option. The first element is title, which indicates the order in which to scan the library. The second element, “statement”, is the starting point for the scan, and the third element is the ending point for the scan.

Extract a report from the archive

If you have defined a user .sdstor file that contains a valid login and password, or you have defined a default login and password in the sdstor.ini file, then you won't need to add a -login option to the scanning commands. If necessary, you can add '-login sampuser/123' to each command, and you will be logged in as the sample user defined above, which has been granted read rights to the sample library.

Note the ID codes from the scan lists. In the examples shown, the ID codes are CB, CC, and CD. Your codes will very likely be different. In the command samples below, use your codes.

The following commands can be used to extract text, laser, and pdf versions of a report to a file. You can also extract to a print device or spooler.

This example will extract a report as plain text to a file:

```
sdstor -lib sample -id cb -o invoice.txt
```

This example will extract a report using PCL5 (HP Laserjet) codes to scale the output to a page, suitable for raw printing on a laserjet printer:

```
sdstor -lib sample -id cb -laser -o invoice.pcl
```

This example will produce a PDF file of the report:

```
sdstor -lib sample -id cb -pdf -o invoice.pdf
```

This example will produce a laser-formatted version and send the output to a spooler. The example shown is for Linux:

```
sdstor -lib sample -id cb -laser | lpr -b -P laserjet
```

This example uses a printer alias named LP, defined in the config.sto file. A printer alias can send output to a spooler, an UnForm job, or a LPTn or UNC device name on Windows.

```
sdstor -lib sample -id cb -laser -o LP
```

Using rule files for automatic capture options

Create a text file called sample.rul in the sdStor directory, using any text editor. The .rul extension is simply a convention; you can use whatever file name you want.

In this file, add the following lines:

```
[ARAging]
detect 50,2,"Detail Aging Report"
title "A/R Aging Report"
```

The [ARAging] line indicates a new rule set. You can have any number of rule sets in a rule file.

The detect line identifies an Aging report. Whenever a job is captured that contains the phrase “Detail Aging Report” at column 50, row 2, then this rule set is applied (assuming, of course, that this rule file is specified by the command line or as the default rule file). The third line sets the title for any such report to “A/R Aging Report”.

Save the file and execute this command:

```
sdstor10 -i sample3.txt -o sdstor.out -lib sample
```

Scan the library and note that the title of the last captured report is “A/R Aging Report”.

Command: **sdstor10 -lib sample -scan id**

| |
|--|
| ID;TITLE;RUNDATE;RUNTIME;PAGECOUNT 000000CB;05/30/97 0005138 000100 000001 Everest Industries Everest In;10/28/2002... 000000CC;STATEMENT SYNERGETIC DATA SYSTEMS INC. 2195 TALON... 000000CD;05/30/97 ADD+ON Manufacturing Company Page 03:40 PM Detail A;10/28/... 000000CH;A/R Aging Report;10/28/2002;16:00:39;3 |
|--|

You can define rule sets to set any of the updateable report properties. See the Automating Report Properties with Rule Files chapter for more information.

Deleting an archive from the sample library

Select an archive ID to delete from the scan list. For example, the above list indicates ID CD is an aging report, which could be deleted, like this:

```
sdstor10 -id cd -lib sample -delete -login admin/12345
```

Note the `-login` option, which specifies the admin user and password defined above. This user has full rights to all libraries, so it can move an archive from the sample library to the trash library. The regular user could not.

You can scan the trash library and see that the archive is now there.

`sdstor10 -lib trash -scan id`

| |
|---|
| ID;TITLE;RUNDATE;RUNTIME;PAGECOUNT 000000CD;05/30/97 ADD+ON Manufacturing Company Page 03:40 PM Detail A;10/28/2002... |
|---|

To permanently delete this report from the trash library, you could enter the following command:

`sdstor10 -id cd -lib trash -delete -login admin/12345`

To transfer the report back to the sample library, you could enter this command:

`sdstor10 -id cd -lib trash -transfer sample -login admin/12345`

Integrating With UnForm

sdStor is designed to use UnForm to produce enhanced report and form output when retrieving reports. This is particularly useful when retrieving forms, such as invoices or purchase orders, as the plain printed data is rarely useful on its own. UnForm's output can be directed to laser printers which support PCL5, or PDF files.

Note that archive formats may differ over time as program changes are made. If this happens, the rule file will need to take into account the possibility of old and new formats for the same report. This may require only some simple conditional additions to a rule set, or it may require duplicating rule sets and adding more detailed detection logic to each rule set copy to distinguish the different formats.

Configuration

Modify the `sdstor.ini` file's `uniform` section. As an example:

```
[uniform]
command=uniform50
dfrule=acme.rul
ar=arreports.rul
```

The `command` value should be the command used to execute UnForm. Note that UnForm must reside on the same system as the sdStor. On Windows, the command will likely be a full path, such as `c:\uniform50\uniform50.exe`.

The rule file should be the standard rule file used by your application. Note that if you wish to generate enhanced PDF reports, or if the sdStor Viewer should invoke UnForm-formatted viewing, then the rule file will need to support PDF output. This generally means that additional image and/or attachment data files must be generated in PDF format. In some cases, copy management will require modification as well.

There is a default rule file set as `dfrule=rulefile`, and there can be library-based rule files. When an archive is retrieved from a given library, sdStor first looks for a rule file specification base on library name. In the example above, if a report is retrieved from a library named "ar", it will use `arreports.rul`. If no matching library rule set is found, then the `dfrule` value is used.

The View Server should also be configured to offer an UnForm viewing option for the sdStor Viewer. Add or modify a line in the `[defaults]` section of the `sdstord.ini` file to read `"uniform=1"`.

Command Line Control

Whenever a command line is used to retrieve a report, by using the `-id` option, it is possible for `sdStor` to execute `UnForm` to produce the final output by simply adding the `-uniform` option to the command line, as in these examples:

```
sdstor10 -id 6c -uniform | lpr -b -P laser
```

```
sdstor10 -id 6c -uniform -pdf -o /tmp/invoice.pdf
```

Note that the default format for an `UnForm` formatted job is `PCL5`, so it must be directed to a laser printer that supports `PCL5` or higher. In addition, when spooling an `UnForm` job (as well as a standard laser-formatted `sdStor` job), the spooler must be invoked with options that turn on binary, raw, or pass-thru mode.

Aliased Printer/Link File Control

`sdStor`, when run under `PRO/5` or with a bundled runtime, supports printer definitions in the `config.sto` file. On Unix, an alias line consists of three elements, like this:

```
alias LP "[uniform50 -f acme.rul | lp -dhp4000 -oraw]"
```

On Windows, a more complex alias is required, which stores `sdStor` output temporarily in a text file, then executes `UnForm` to process the text file and send the output to a device. See your `UnForm` documentation for more details about `UnForm` commands.

```
alias P1 c:/temp/p1.txt "UnForm alias"  
o_create,spcols=255,execoff="c:/unform50/unform50.exe -ix c:/temp/p1.txt -f  
acme.rul -o LPT2"
```

On `ProvideX`, you can define a printer link file using one of the `UnForm` print drivers. The link file can be named with a full path argument to the `-o` option, or can be placed in the directory that will be the "current" directory when the user executes the `sdStor` command line.

When sending `sdStor` output to an `UnForm` printer, do not use the `-uniform` option. This would result in `sdStor` output being binary `UnForm` output, in turn going to `UnForm` as input.