http://ads.ahds.ac.uk/

ADS/AHDS Archaeology Ingest manual

Status: Fourth draft
Version: 2
Authors: Tony Austin and Jenny Mitcham
Date: 19th April 2006

Section numbers in this document relate to those used in the AHDS Archive Ingest Procedure Framework: HS Preservation Procedures Manual, working draft 1.3 prepared by Raivo Ruusalepp, Estonian Business Archives Ltd, December 2002/January 2003. Please refer to this document for more detailed information.

4.2 Accession / Submission

The process of deposition of data is administered using a AHDS wide Collections Management System (CMS) which is accessible from the following url: http://muninn.york.ac.uk/cms/ and uses Java Server Pages to interact with an Oracle database. This is replaces the original ADS system which was implemented in Microsoft Access and managed via form-based interfaces. The CMS has four underlying databases concerned with four distinct areas; data producer details, project tracking, post deposition management of collections and user services.

As a new Submission Information Package (SIP) is received by the ADS/AHDS Archaeology it is a priority to get data stored safely on our systems and issue the data producer with a written confirmation of receipt. Accessioning will normally be carried out within 5 working days of receipt of data.

Problems with any of the processes listed below should be noted in the CMS.

An accessions checklist is available to guide curatorial staff through the accession process.

4.2.1 Data transfer session

Data is accepted on 3.5 inch double-density or high-density floppy disks, CD-ROM, DVD, Zip discs, over email, external drives or via FTP or web upload.

Should media be damaged or a transfer session interrupted ask data producer to resubmit.

For data rescue purposes we may also able to read data on 5.25 inch disks though this is not a format we generally encourage.

.

4.2.2 Update metadata

Date of receipt of data is to be recorded in the tracking module of the CMS. This will allow the data accessioning process to begin.

4.2.3 Virus check

Before storing the data Sophos Anti-virus software is run on the SIP to ensure it is virus free. If a virus is found, this must be removed either by disinfecting the relevant file, or requesting that the data producer provides a 'clean' copy. Further processing of the data files must not go ahead until the data is known to be free of viruses.

Virus checking will highlight any files that are password protected. Data producers should be asked to resubmit copies of any files that are password protected.

4.2.4 Media and file readability check

If the SIP consists of only a small number of files these are opened to ensure they are readable and not corrupt. Unless batch processing is an option only a random sample of each format present in a large archive will be checked at this stage. More thorough checks will be carried out in the data processing stage. The data producer should be contacted regarding any corrupt files and asked to resubmit.

Though we prefer to receive unencoded data, we can accept UUENCODED files.

If data is compressed, we ask our data producers to use one of the following formats: GNU (.gz), Pkzip, Stuffit, TAR, Unix compressed files (.Z), Zip. If a different compression is used, we may need to contact the data producer and ask for an uncompressed version.

4.2.5 Data resource completeness / integrity check

The SIP should be checked against any relevant documentation such as file lists supplied by the data producer. The numbers of files, file names and formats should be checked here. The purpose of these initial checks is to highlight any missing files or major discrepancies between the delivered data and the documentation. Any discrepancies should be queried with the data producer (see also 4.2.8).

Use DOS based checksum utility FastSum to create checksums for the data before making a copy of it. Once data transfer has occurred, these checksums can be compared with new checksums created to ensure the data transfer has been successful (see 4.2.8).

FastSum will need to be installed locally before it can be run. The .exe file is available within the software_archive directory on Archads1.

Full documentation of the FastSum utility can be found on your local drive within \Program Files\FastSum\ after the software has been installed.

Open a command prompt from Windows and navigate to C:\ - then type
FSUM "D:\" C:\filename.txt /R
This will run checksums on the D drive, /R means it will work through all directories on the CD. The results will be stored in a text file named filename.txt.

4.2.6 Documentation completeness check

ADS/AHDS Archaeology Data Procedures documentation should be consulted to locate any gaps in supplied documentation. Guidelines exist for each broad category of file we are likely to receive.

Data producers should be requested to provide any missing documentation necessary for the processing and archiving of their data. Requests can be sent direct to the data producer or can go via the Collections Development Manager.

4.2.7 Copy to processing area

Data is copied on to an external hard drive. This data is synchronised on a weekly basis with a hard drive that is stored off site. Where data has been sent by e-mail or ftp, original copies should be retained until this data synchronisation has taken place.

All data is copied into an appropriately named root directory within the ADS_Preservation directory. This directory will be named according to AHDS guidelines using the structure *arch-{ADS collection number}-{edition number}*.

Within this new directory, the SIP is placed in an appropriately named directory (*yyyy-mm-dd*) under another directory called *original*. Any directory structure within the original data is mirrored in the *original* directory. The data in this directory remains unedited and provides a reference copy of the original SIP.

Any metadata or documentation about the collection included with the SIP must be duplicated in the *admin* directory for that collection.

For more detailed information on the directory structure used for ingest, refer to *AHDS Repository Operations* by Hamish James and Gareth Knight (http://atticus.ahds.ac.uk/local/CollectionsWG/repository_operations_d2.doc last accessed 19th April 2006).

The directory _*template* which can be found on the external hard drive can be copied, renamed and used as a template for setting up the correct data structure for a new collection.

Once transfer to the hard drive is complete, original data media and associated paperwork are stored in the collections filing cabinet ordered by collection number. Other documentation relating to the project is filed by the Collections Development Manager.

4.2.8 Authenticate original version

After data transfer the FastSum utility must be re-run on the files in their new location. This is necessary to ensure that the files within the *original* directory are identical to those submitted. Checksums for files on remote directories can be created using the Windows command line FastSum utility if the ADS filestore is mounted locally using WebDrive.

Compare these checksums with those created under 4.2.5. The files are not normally identical as the checksums are not carried out in the same order each time the procedure is run. It will normally be necessary to compare the two files within a database application to check for records in the first checksum file that do not have an identical record within the second checksum file.

Where a problem is identified the data transfer session must be repeated until checksums match up.

An alternative to this process of comparing checksums would be to use the FolderMatch software to run a comparison on two directories. Comparisons can be made in a number of different ways, by filename, size and date/time or more accurately by SHA-1 message digest. This is a different checksum algorithm to the MD5 used by FastSum. It is the most reliable method of ensuring that files have not changed during transfer.

Once this process has been carried out, the only editing of the *original* directory that is permissible is the removal of spaces from filenames. Many people use a Windows or Mac OS where spaces in filenames are legal. As our data servers run on UNIX, spaces in filenames can cause problems when trying to access the files. A Perl script is available which runs through directories and filenames and replaces all spaces with the underscore character. The script is called replace_spaces.pl and can be found on our server Mnemoyne at /export/home/people/ads/ collections/scripts/. This script can be run on the *original* directory and will run through its subdirectories by default. It must only be run after the authenticity of the files is assured by the use of checksums described above.

It may also be necessary at this point to amend filenames which contain accented characters. There is currently no script available to carry out this operation.

Finally, an MD5 checksum command must be run on the *original* directory to create a checksum to be stored alongside the data. This can be done using FastSum or md5batch.sh which is located in the *scripts* directory on Mnemoyne under the collections user. The file that this command generates must be stored in the *admin* directory of the collection (see Appendix A for further information on running the md5batch.sh checksum utility).

4.3 Administration and metadata management

4.3.1 Check all forms

The Collections Development Manager checks that the Deposit Agreement/ Licence Form has been signed by the data producer and updates the project tracking database. One copy of the Deposit Agreement/ Licence Form is retained by the data producer and another is held by the ADS/AHDS Archaeology. Only when the Deposit Agreement/ Licence Form has been signed will the processing of the SIP proceed.

Licences are scanned on receipt and are stored in the *licences* directory on archads1. Licences should be scanned to the specification provided by the AHDS.

During the accessioning process a scanned copy of the signed licence agreement must be moved into the *admin* directory for the collection and named *licence.tif* or *licence.pdf*.

4.3.2 Check copyright and confidentiality clearance

The Collections Development Manager checks that any associated copyright and confidentiality clearance issues have been appropriately addressed by the data producer and noted on the project tracking database. Should there be any unresolved copyright issues the processing of the submission stops until relevant copyright clearances have been obtained.

When necessary, the Collections Development Manager refers to Part IV of Schedule 8 of the 1998 Data Protection Act, which lists conditions under which collected data are exempt from Data Protection clauses for the purposes of research, and acts accordingly.

4.3.3 Update metadata databases

Once a deposit licence for a collection has been signed and received by the Collections Development Manager it is recorded in the CMS.

How the SIP is recorded in the CMS depends on the nature of the deposit.

If the SIP is data for a new collection for which we have not received any other data, the accessioning process will need to be followed through from the tracking module of the CMS. Details of the project, funder, contacts and deposit licence should already be entered into the system. If there are no details available, speak to the Collections Development Manager before continuing. If the project record exists in the tracking module, enter the date the SIP was received and click the 'accession' button to enter fuller details of the deposit.

If the data is an addition to an existing collection or a reload of data already held by the ADS the process may be a little different. (watch this space – will be completed once accessions module of new CMS is implemented)

Some examples to explain when an SIP is a new collection and when a new accession for existing collection:

- **Updated database from an SMR that is already held by ADS:** This will be a new accession for the existing collection
- **A new volume of an archaeological journal where other volumes are already held by ADS:** This will be a new accession for the existing collection
- **An extra file that we have requested as part of the accessioning process (for example we may ask them to resubmit a piece of data or request something that is missing)** This will be added to the original accession though it will sit in a directory within /*original*/ named with the date received
- **A grey literature report submitted through the OASIS form:** Each unit has their own collection for their OASIS reports. Each time more reports are uploaded into the grey literature library for this unit (reports are uploaded on a monthly basis) this will produce a new accession for this collection

Ingest should not proceed unless the existence of a deposit licence is registered. If there appears to be no signed licence for the data, query this with the Collections Development Manager.

4.3.4 Scan paper documentation

Most documentation arrives in digital form, on the occasion where only paper copies are provided and we are subsequently unable to get a digital version from the data producer, paper documentation will be scanned at an appropriate resolution as determined by the AHDS guidelines contained within the *AHDS Repository Operations* documentation. This should be stored in the *admin* directory for the collection alongside any other documentation in digital form.

4.4 Quality assurance

The ADS/AHDS Archaeology endeavours to undertake validation to the extent of the documentation supplied. It may be necessary to contact data producers if problems are encountered during validation.

4.4.1 Review data and prepare validation

Formulate a validation plan based on the documentation, file types and objects (such as databases) supplied. AHDS Guides to Good Practice, Preservation manuals and ADS/AHDS Archaeology Occasional Papers can guide this process.

Guides: http://ads.ahds.ac.uk/project/goodguides/g2gp.html

Preservation manuals: http://atticus.ahds.ac.uk/wiki/Wiki.jsp?page=PresHandbooksColls

Occasional papers: http://ads.ahds.ac.uk/project/userinfo/occ_pap.html

Data Procedures documentation: (available through the ADS staff web pages)

A record of inconsistencies and problems during validation can be stored in a notes field in the Collections Management System.

4.4.2 Consistency checks

Basic checks of data against accompanying documentation should already have taken place under section 4.2.5

The nature and complexity of consistency checks depends on the data type under scrutiny. Databases, GIS and spreadsheets would usually require more checking than raster image files for example.

Here are some examples of the types of checks which should be carried out (taken from the AHDS Archive Ingest Procedure Framework Draft 1.3):

- Check that digital resources and their items adhere to the relevant formal definitions of their structure (e.g., an XML document conforms to its XML schema, a relational database conforms to its SQL schema, an image conforms to its stated image format – dpi, colour depth, compression, etc.).
- Image compression algorithm, dimensions, orientation, resolution, colour space, etc. correspond to the values stated in documentation.
- Digital audio compression algorithm, length of the recording, sampling frequency, bit rate, etc. correspond to the values stated in documentation.
- Digital video compression algorithm, length/duration of the recording, codec structure, frame rate, sound format, etc. correspond to the values stated in documentation.
- Linkages and dependencies between items within a particular type of digital resource should be checked for correctness (e.g., in a database, foreign keys having a matching primary key; in a spreadsheet, formulas refer to correct cells, etc.).
- Linkages and dependencies to other digital resources are correct (e.g., hyperlinks point to a currently valid URL, details of published works in a bibliography are correct, etc.).

- Items within a digital resource adhere to the relevant definition (e.g., a numeric field in a database contains a number, text strings do not exceed a stated maximum length, etc.).
- Items within a digital resource contain ‘sensible’ values that do not contradict relevant logical assumptions (e.g., age of a person should not be less than 0) and subject/resource type specific concerns.
- Documents (word processor files) should be checked for changes or errors in footnotes, tables of contents, links, auto-fields and formatting that may hinder the later use of the data resource.
- GIS, CAD and virtual reality data resources may require domain- or research area specific consistency checks to be applied (e.g., scale of different layers in a GIS, level of precision and sufficiency of co-ordinates in a CAD and VR data, etc.).
- Simple data types (numbers, text strings, dates, etc.) are not truncated, restricted in range, formatted or otherwise defined in a potentially confusing or ambiguous way (e.g., dates contain four digits for the century, date format, memo fields in a database do not contain embedded end-of-lines, etc.).
- Coded data must be checked that the data have been consistently assigned the documented code.
- Any codes that are used in data must be used consistently and according to the specified coding rules.
- Standardised data has been standardised consistently and according to specified rules or a recognised schema for the standardisation.
- Exceptions to particular standards, coding schemes, formats, etc. are documented and justified in the documentation for the data collection.

Where problems with any of the data are highlighted, the data producer should be contacted, informed of the problem and encouraged to re-submit. Altering or editing the data within the *original* directory should only be carried out with permission from the data producer and any correspondence relating to this should be stored in the *admin* directory of the collection. This should only be carried out where the data producer refuses to re-submit.

4.4.3 Metadata update

The information added to the Collections Management Database must include the delivery medium, the number of files and their file type (and version if appropriate) as well as the storage location on our data server.

Particularly useful for large deposits of data, there is a Perl script that runs through files in a directory and counts the number of files of each file type. The script is called list_count_files.pl and can be found on our server Mnemoyne at /export/home/people/ads/collections/scripts/. When this script is run, any unrecognised file types or files with no file extension would be highlighted and the data producer could be queried if necessary.

Once a digital resource has been accessioned, a receipt must be created to send to the data producer including the project title, the date of receipt by the ADS/AHDS Archaeology, and the number of files and types of files received.

This receipt can be automatically generated by the database as part of the accessioning process. A copy of this receipt should also be given to the Collections Development Manager for filing in the project file.

4.5 Creating the preservation version

AHDS Guides to Good Practice, ADS/AHDS Archaeology Occasional Papers and Preservation manuals can guide preservation strategies.

Guides: http://ads.ahds.ac.uk/project/goodguides/g2gp.html [last accessed 20 May 2005]

Preservation manuals: http://atticus.ahds.ac.uk/wiki/Wiki.jsp?page=PresHandbooksColls [last accessed 20 May 2005]

Occasional papers: http://ads.ahds.ac.uk/project/userinfo/occ_pap.html [last accessed 20 May 2005]

ADS Data procedures documentation (available from the ADS staff pages)

The taxonomy of data types and file formats in the AHDS collection, working draft 3.1 prepared by Ravio Ruusalepp, August-October 2002. A digital version of this document is available at the following url: http://atticus.ahds.ac.uk/local/preservationconsultancy/D2_Taxonomy_final.pdf.

4.5.1 Choosing the preservation method

Create a local copy of the files from the *original* directory so there is no chance of altering the original data. This temporary copy can be deleted once work on the files is complete. Where appropriate use checksums to ensure you are dealing with the same data that is in the *original* directory.

Refer to the Collections Management System to see which file types and versions have been deposited. Using this and the files themselves identify suitable preservation formats for each of them. In carrying out this step it is important to consider the significant properties of each of the files and ensure that these will be preserved within the new format.

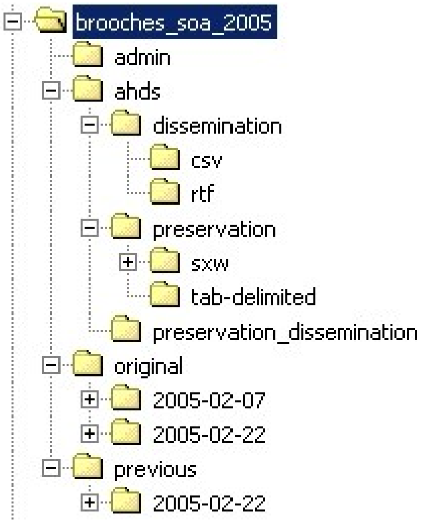
Refer to the documents listed above to inform decisions on a suitable preservation format.

4.5.2 Develop a conversion plan

If the conversion into a preservation format is a complex procedure, map out each of the steps to be taken and check that we have the software and expertise to carry out this work. Where appropriate the AHDS executive or other AHDS subject centres may be able to help or advise.

4.5.3 Convert the files

In the relevant directory on our data server (see 4.2.7) there should be a series of directories alongside the *original* folder. Files converted for preservation purposes should be stored within the *preservation* directory under *ahds* unless the same file will also be used for dissemination, in which case it should be stored in the *preservation_dissemination* folder. Directories within these folders should be named according to the final preservation file formats identified in 4.5.1. For example, in the file structure below there are separate directories to contain openOffice document (sxw) and tab delimited data files.



```
brooches_soa_2005
    admin
    ahds
        dissemination
            csv
            rtf
        preservation
            sxw
            tab-delimited
        preservation_dissemination
    original
        2005-02-07
        2005-02-22
    previous
        2005-02-22
```

For more information on this file structure see *AHDS Repository Operations* by Hamish James and Gareth Knight.

Following instructions contained within the Data Procedures documents, copy and convert the deposited files as necessary until every file supplied is in the correct format and location.

4.5.4 Validate file conversion

Ensure that the conversion of the digital resource has been successful and that the significant properties of each file remain unchanged.

4.5.5 Authenticate the preservation version

Once preservation versions of files are validated, authentication information must be attached to them. The MD5 checksum routine should be run on the whole directory to create a digital signature for each file. Once again the resulting checksum file should be moved over to the data storage area and placed in the *admin* directory of the relevant collection. For collections that are to be transferred to the AHDS, this final checksum file should be renamed *checksum.txt*.

4.5.6 Metadata update

Once the conversion plan has been successfully implemented, the Collections Management System is updated to include details of these conversions. This will include detailed preservation metadata on the conversion process, including methodology and software and hardware used which was previously recorded in the Generic Preservation Metadata Database tool. Full documentation for this preservation metadata can be found on the AHDS Wiki.

Once completed, this data record may be saved as an XML document and should also be stored within the *admin* directory for the collection. (Is this still going to be the case????)

NOTE: The following sections apply to all projects that are AHRC-funded, plus other HE projects as specifically approved by ADS Director.

4.6 Prepare the Archival Information Package

If all the processes above have been followed, the collection should now be in a fit state for submission to the AHDS Executive. Check the *AHDS Repository Operations* documentation to ensure the recommended directory structure has been maintained, and required metadata and checksums are located in the *admin* directory of the collection.

4.7 Submit the Archival Information Package for Preservation

Once an Archival Information Package (AIP) has been prepared for transfer it should be submitted to the AHDS Executive.

The AIP will consist of the full directory structure for the collection, including the contents of the *original*, *admin*, *ahds*, and *previous* directories. The AIP can be zipped into a single file before transfer if appropriate but the zipped archive should maintain the original directory structure.

When transferring the AIP the root directory for the archive must be renamed. Each collection should be given a unique name in the following format: arch-{ADS collection number}-{edition number}.

For full instructions on transferring collections to the AHDS repository see *AHDS Repository Operations* documentation.

After the AIP has been transferred, a copy of the collection should be stored on the ADS external hard drive. The AHDS Executive will not take full responsibility for the data until checks have been carried out and confirmation sent.

The only data that should be stored on */adsdata/* directory on Mnemoyne are those within the */ahds/dissemination/* and *ahds/preservation-dissemination/* directories.

Appendix A – md5 checksums

Data integrity

The Message Digest number 5 (md5) value for a file is a 128 bit integer similar in some ways to conventional 16 bit and 32 bit checksums. Open source software for generating md5 signatures is available for most platforms. An md5 signature for a file is effectively unique. The chances of another file having the same md5 value is extremely unlikely to the extent that versions of a file which are the same size in bytes can be distinguished. However, md5 is consistent. If nothing has changed running md5 on a file should generate the same value whatever the platform. Thus files corrupted or accidentally renamed during transfer can be identified. It is also thought that it is currently impossible to fake an md5 value.

Generating file signatures

The md5 command has been installed on Mnemoyne. Further it has been built into a shell script (see below) that also generates other fixity values from the system either for a specified directory (including sub directories) or, if none stated, using the current directory. The script also formats the data and adds necessary header information for using with SQL*Loader. The script, md5batch.sh, has been moved to /usr/local/bin on Mnemoyne and made executable. For the script to be effective you need to log in as collections and su to root otherwise file ownership may interfere. Move to the scripts directory and type:

```
sh md5batch.sh
```

You will then be prompted to type the directory that you want to run the script on.

```
login: collections
Password: ********

Last login: Mon May 13 17:33:39 from ads4.york.ac.uk
tcsh: using dumb terminal settings.

Collections>> su
Password: ***********
tcsh: using dumb terminal settings.
```

To run the script

```
Collections>> sh md5batch.sh
Enter directory path to be processed
For example, /data/data3/cleobury/
```

```
Current location will be used if no
directory is specified

/data/data5/cbaresrep/tif/001

Processing /data/data5/cbaresrep/tif/001
finished processing /data/data5/cbaresrep/tif/001
Oracle import file is  MD5SUMS.20020521141723

Collections>> cat MD5SUMS.20020521141723
LOAD DATA
INFILE *
APPEND
INTO TABLE MD5CHECKS_PRES
FIELDS TERMINATED BY '|' OPTIONALLY ENCLOSED BY '"'
TRAILING NULLCOLS
(ORGANISATION, ADS_DATE DATE "DD-MM-YYYY", ADS_TIME,
PATH_FILE, SIZE_BYTES, MD5_
SIG)
BEGINDATA
ADS | 21/05/2002 | 14:17:23 | /data/data5/cbaresrep/tif/001/aa000001.tif |
21365 | b25da9fcec6f88f2a34d84a325451ca5
ADS | 21/05/2002 | 14:17:24 | /data/data5/cbaresrep/tif/001/aa000002.tif |
52413 | 7c50fd2be947857b2394027f364c7636
ADS | 21/05/2002 | 14:17:24 | /data/data5/cbaresrep/tif/001/aa000003.tif |
8467 | 837545a16429ab0317cabef4179b5fbe
ADS | 21/05/2002 | 14:17:24 | /data/data5/cbaresrep/tif/001/ab000004.tif |
132757 | 9b7dc4f744e51d352aabdc4b3cb3b601
and so on……
```

This file describes an Oracle table (with some fields specific to UKDA) that has been created to hold this data.

```
SQL> desc  MD5CHECKS_PRES
 Name                          Null?    Type
 ----------------------------- -------- ----
 ADS_DATE                                 DATE
 ADS_TIME                                 VARCHAR2(12)
 UKDA_MOD_DATE                          VARCHAR2(12)
 UKDA_CREATE_DATE                       VARCHAR2(12)
 UKDA_ACCESS_DATE                       VARCHAR2(12)
 PATH_FILE                               VARCHAR2(500)
 SIZE_BYTES                              NUMBER(10)
 MD5_SIG                                 VARCHAR2(50)
 UKDA_ATTR1                             VARCHAR2(20)
 UKDA_ATTR2                             VARCHAR2(20)
 UKDA_ATTR3                             VARCHAR2(20)
 UKDA_ATTR4                             VARCHAR2(20)
 UKDA_ATTR5                             VARCHAR2(20)
 ORGANISATION                           VARCHAR2(20)
```

The data can be loaded directly…

```
Collections>> sqlldr userid=ads/*** control= MD5SUMS. 20020521141723
log= MD5SUMS. 20020521141723.log
```

The log file can be checked to make sure data loaded correctly. If so various data and log files can be deleted. Could automate Oracle data loading from shell script but there are situations where intervention is necessary. For example, when the default, current location, is used fixity values will be created for the shell script which by definition is in this directory. This would need editing from the data. Sub directories may also create values which will need removing.

As well as generating comparatives for the ftp process the Oracle table provides file level metadata for management procedures.

md5batch.sh script

The script code is listed below:

```
#!/sbin/sh

# Tony Austin 10 May 2002
# md5barch.sh: creates MD5 signatures for selected directory
# also records file path +name, bytes, date, time into a
# delimited text file, MD5SUMS.<date>

#environment
umask 022
PATH=/usr/bin

# check that user is root
user=`id | nawk -F\( '{ print substr($2, 1, index($2, ")") - 1) }'`
if test $user != "root" ; then
   echo >&2 "Error: $0 Run as root or files may be missed."
   exit 0
fi

#get directory for processing

echo "Enter directory path to be processed"
echo "For example, /data/data3/cleobury/"
echo "Current location will be used if no"
echo "directory is specified"
read DIRECTORY
if [ "$DIRECTORY" = "" ]; then
DIRECTORY="`pwd`"
fi
```

```
echo ""
echo "Processing $DIRECTORY"

#set up variables and files

DATE=`date +%EY%m%d%OH%OM%S`
MD5=/usr/local/bin/md5
FILELIST="`find ${DIRECTORY} -type f `"
OUTFILE="MD5SUMS"
export OUTFILE
OUTFILE="${OUTFILE}.${DATE}"

GetSigs() {
if [ "$1" = "" ]; then
        echo "ERROR - no files in directory"
        return
else
  FILELIST2="${1}"
fi
rm -f ${OUTFILE}
touch ${OUTFILE}
#loop through file list
for files in ${FILELIST2}
do
   if [ -f ${files} ]; then
        ORGANISATION="ADS"
        DATE2=`date +%d/%m/%EY`
        TIME=`date +%OH:%OM:%S`
        BYTES="`wc -c $files | cut -d"/" -f1`"
        MD5SIG="`$MD5 $files | cut -d"=" -f2`"
        PATH_STR="`$MD5 $files | cut -d"=" -f1`"
        PATH_STR2="`echo $PATH_STR | cut -d\) -f1`"
        PATH_FILE="`echo $PATH_STR2 | cut -d\( -f2`"
        echo            "$ORGANISATION | $DATE2 | $TIME | $PATH_FILE |
$BYTES | $MD5
SIG" >> ${OUTFILE}
   fi
done

}

GetSigs "${FILELIST}"

# add sqlldr header to the data
cat sqlldr_md5_header.txt $OUTFILE > md5.tmp
mv md5.tmp $OUTFILE

# set file attributes for collections user
chown collections:ads $OUTFILE
chmod 751 $OUTFILE
```

```
echo "finished processing $DIRECTORY"
echo "Oracle import file is $OUTFILE"
```