ASCix : A Simple Cataloguer for Heritage Data

Steve Nickerson steve@icomos.org Rob Swan ascix@sympatico.ca
C A R T : Computer Aided Recording Tools
501-99 Holland Avenue, Ottawa, Ontario, Canada K1Y 0Y1

KEYWORDS: Data Structure, Documentation, Internet/Web, Management,

ABSTRACT

Heritage Data must be accessible to the world at large and it must be maintained for very long periods, hopefully forever. Reconciling this requirement with the electronic formats in which our data is being created and stored has been the subject of much debate but few solutions.

Previously a Heritage Record consisted mostly of texts and photographs and, if the field notes were reasonably well maintained, anybody understanding the language could figure it out. Today a lot of this data is created in electronic formats requiring specific hardware and software to access it. The files can be preserved by constant renewal, itself a daunting (but doable) task, but what is the advantage of preserving the electron stream if the tools to make sense of it are lost?

The model we want to emulate is an ancient inscription where it doesn't matter what sort of stone was used or what kind of chisel. All that should be needed to decipher the data is knowledge. We thought that the closest we could get an electronic approximation of this ideal would be a data set where any researcher able to get a directory listing could decipher the organization of the data with only a text editor.

ASCix is A Simple Cataloguer which prepares such a data set. It grew out of a much more (much too) complicated database application for archaeology but has been refined into a tool that:

1. Renames each file to a predetermined standard so that a sorted directory listing would provide some idea of the contents.
2. Creates flat files describing the catalogued files, the logic according to which they were prepared and the types of software needed to open them.

One or two minutes per file is all it takes and the data will be in a package decipherable by any computer literate researcher, even if the project is abandoned before publication.
This paper will provide a brief introduction to the program with examples of its use and the output resulting from a recent course in Heritage Recording where students from around the world used this as their primary tool while recording Santa Cecilia in Trastevere, an 18th century rebuilding of a 9th century church in Rome.

There are two main types of data in the field of Heritage

1. **Presentation Data**, this is best characterized as being a small data set of interest to a large number of people and is represented by the idea of the museum. It takes a small sub-set of a collection and assembles it into a presentation designed to tell a story to a large number of visitors.

2. **Record Data**, this is the opposite, a large data set in which almost nobody is interested, but from which the presentation data is gleaned. This is the source material, gathered in the field or from other archives and assembling it is the most important component in the understanding of a resource. It is also the most difficult, time consuming and expensive. The people who gather it are almost always working under constraints of time and resources and every effort must be made to make it as easy as possible for them to get their data into the system (any system).

Long Term Storage

Another way that heritage data differs from that generally addressed in data base management models is that this record must be accessible to the world at large and it must be maintained for very long periods (forever). This precludes the use of proprietary formats. In fact it is best if no software at all is necessary to access the data.

A logical extension of this principle would suggest that non-proprietary formats be used for the files to be catalogued. Word Processing documents can be saved as .TXT, Spreadsheets and Database files as .CSVs and CAD files as .DXF. Where significant data loss accompanies such a conversion both files can be catalogued (just in case posterity has the appropriate software).

If you are not going use software to organize your data it follows that the only way to keep everything in order is to use the files themselves to do it. This can be accomplished if each file has a unique name such that, without any software the relationships between it and other files as well as to references in the texts can be discerned. Two simple questions can satisfy this requirement: **What is it**? (a wall, coin, bone, book, etc.) and **Which is it**? (Usually a number unique for that "what")

For example: An object named "what_001" exists and is mentioned in the texts and tables of a record. Photographs of the object would be called "what_001*.jpg", the drawings "what_001*.tif" and a text description of it "what_001.txt". There may also be a “what.TXT” and a "what.CSV" which would contain information on all objects of the type "what".

This project started with two main goals:

One was to create a data set that can be deciphered by researchers other than those that created it.

The other was to make it **easy** to create and maintain such an electronic data set.

There are four components to the first goal:

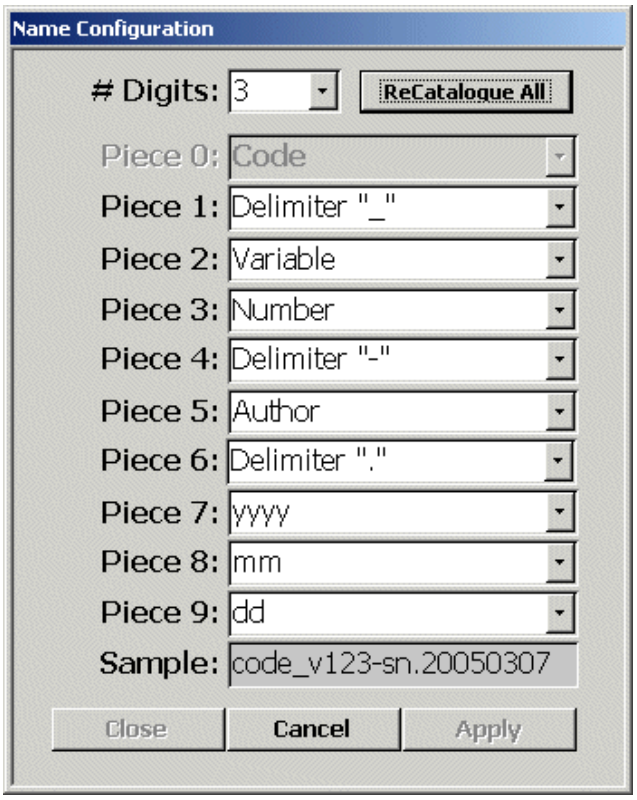
1. Read Me Files
2. Structured File Names
3. Shadow Files
4. Comma Separated (.CSV) Files

Read Me Files

The READ-ME files offer a brief explanation of the organization of the data set. There is a text file that describes the logic behind the data set and a comma separated file showing the rules according to which it was created (the configuration file tabulated for greater readability). The actual file names are written [READ-ME] (.TXT & .CSV) so as to be the first items in a directory listing sorted by name.

Structured File Names

To maintain a directory where every file has a name which, in itself, provides some basic information about its contents a recipe is defined that determines the name of each catalogued file.



File Name Definition Tool

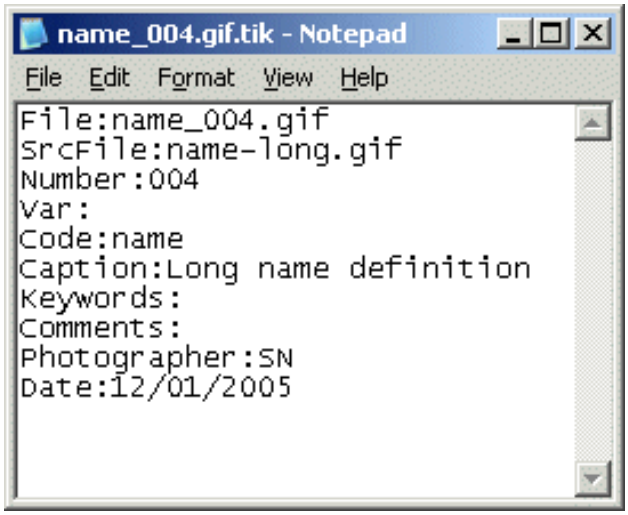
The components of the name can include a code, a date and an author ID. These, along with user defined variables and delimiters, are strung together to create file names that are both unique and completely consistent for all the files in the catalogue.

File names can be quite complicated, as shown in the example above, or as simple as a Code and a Number, which would give a catalogued file a name of “code001", (though a delimiter seems always to be a good idea “code_001"). It is possible to change the file name recipe at any time as long as the change doesn’t cause naming conflicts. For instance a digit can be added if you run out of numbers or a field can be added..

The file naming formula is described in the ReadMe file which heads the directory listing of the catalogue.

Shadow Files

For each file catalogued there is a small text file written. It has exactly the same name as the catalogued file but with an added extension (so that it comes directly below the file it describes in a directory listing sorted by name) and it contains the complete contents of the data base record for the catalogued file.



Shadow file being edited in Notepad

The shadow files ARE the database and a researcher looking at a directory listing can open any file then open its shadow for a description.

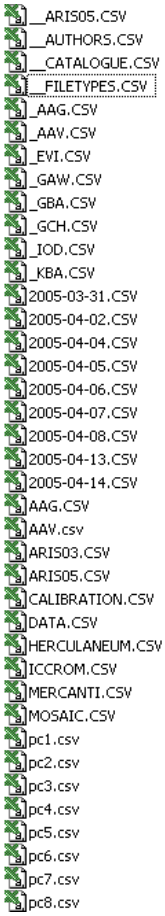
When ASCix opens it reads them all and creates a table which it then compares with the CSV files created during the previous session to see if the data is intact. Currently discrepancies are simply reported but future development will provide tools to recreate missing or damaged shadow or configuration files.

Everything known about the file is contained in its shadow so as long as the the file is kept with its shadow that piece of the database can be rebuilt. To archive or share your data you need only copy the catalogue directory and to combine data sets with the same naming recipe you need only copy one catalogued directory into the other.

Files can be given names with a distinct structure in many different ways, we have done it manually for years (Nickerson, S. 1999a). But it is tedious and easy to make mistakes. ASCix simply helps create this structured data set and helps to make sense of it. The point is to have your files organized, at all times, in such a way as to make all the information is available to someone without ASCix or any other data management software (only knowledge is required).

Comma Separated (.CSV) Files

The comma separated files are compilations of the catalogue data organized in different ways. When a directory is sorted by name these files will be either: at the top if they refer to the whole data set or at the beginning of the group of files they describe.



CSV files preceded by two underscores "__" refer to the whole catalogue.

__AUTHORS is a listing of all the authors keyed on their user ID and showing their names and contact information

__FILETYPES is a listing of all the file types registered keyed on the extension. It lists the type of file (image, text, CAD, spreadsheet, etc.), the full path to the executable, and can include comments on why it was used.

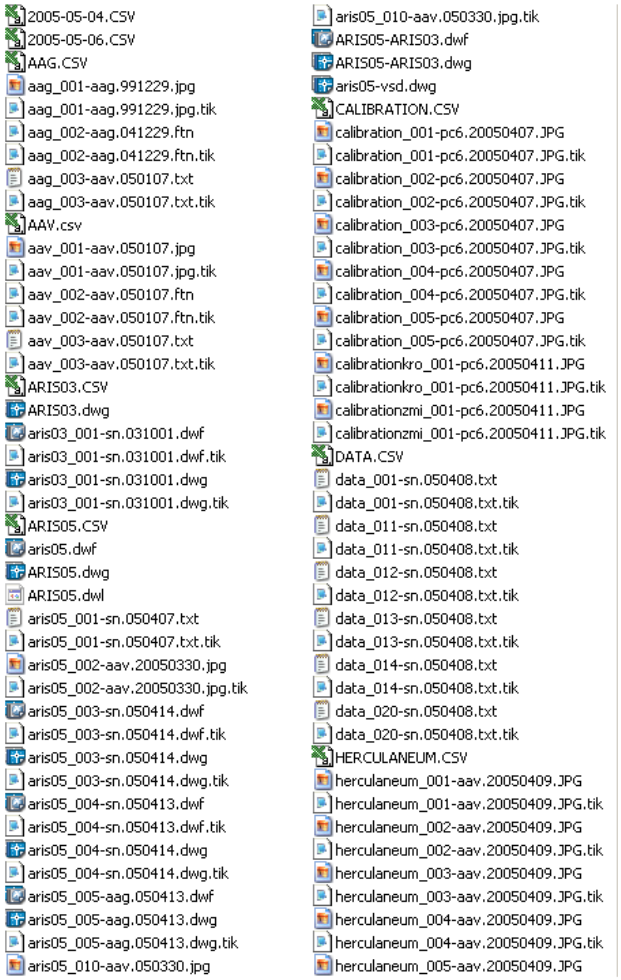
__CATALOGUE is the complete database showing all fields. A search for any word or phrase when editing this file will find it anywhere in the data set.

__catalogueName (in this case __ARIS05) is a reduction of __CATALOGUE showing only the most pertinent fields.

CSV files preceded by a single underscore "_" refer to a user ID

CSV files named for a date will list all the files catalogued for that date.

The rest refer to the codes and list the files catalogued with that code. In a directory sorted by name they will appear scattered throughout the listing acting as a sort of header to the files themselves.



ASCix adds nothing proprietary to the data set and it is hoped that given the Shadow Files and the Reports along with the files themselves any scholar, using whatever software they prefer, should be able to make sense of the data and/or import it into their own environment for further study and analysis.

Using ASCix

Whether we have succeeded in our primary goal of creating a data set accessible to any computer literate researcher is for others to decide and the question of the long term viability of this approach only time will tell. Our second goal, that the interface be easy enough to use that the people in the field will keep their data up to date, needs to be discussed in the context of some field testing, but first a quick overview of what is involved in setting up and using this software.

There are two things you have to do before you start:

1. **Define your Directories**
2. **Define your File Naming Recipe**

Step 1: Defining the Directories

You need to tell the system three things up front:

1. The directory where your source data can be found (this will be empty when you are up to date)
2. The directory where you want the catalogue (this will contain the renamed files, shadows and CSV files)

3. The directories where the source data is to be stored after cataloguing (there are two, Used and Rejected). All your originals are saved whether you wanted them for the catalogue or not(though they can be deleted at any time).

The Codes, userIDs, Keywords and Filetypes can be defined in advance or on the fly.

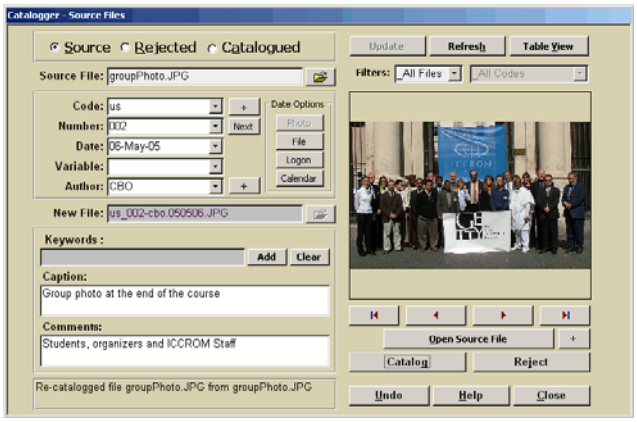
Step 2: Define a File Naming Recipe

This is done through the interface shown earlier. The definition can range from simple to complex depending on your needs and it can be changed later if required so long as the changes don't create conflicts in the file names.

With these two steps completed, and with some files in the Source directory you are ready to catalogue.

This involves a minimum of 3 steps:

1. Select a file
2. Reject it or Select a Code
* A number will be supplied but can be changed as long as it doesn't create a file name conflict
* The date is taken from the file but can be changed
* The author defaults to the current User
** Optionally you can: Add a variable, Select Keywords and Enter a Caption and/or Comments
3. Select Catalogue



ASCix Cataloguer Interface

Of course there can be more to it than that. You can browse through the files having filtered them in various ways. You can open and edit them (using the software you defined in Filetypes). You can browse and catalogue previously rejected files or re-catalogue / un-catalogue files already indexed.

It is **easy** to use, but is it **easy enough** to get field workers to use it?

It should be if they have an appropriate allegiance to the principle of data management. For instance, an important feature of a structured system is that, if the data is ALWAYS kept in a such a format, a discontinuity is less of a problem. How much material has been lost because the excavators never get around to publishing and no one else can decipher their notes?

But just in case they waver in their loyalty to that ideal there is a reward for cataloguing.

Structured file names, as well as providing some meaning for future researchers have the added advantage of allowing automatic processing using tools like CartHtml (Nickerson, S. 1999b)which generates web pages based entirely on file names. Such automation means that you can use and/or distribute your data almost immediately to help make decisions in the field, or to assess its completeness (instead of waiting until months later to find you are missing some crucial piece of information)

Web Pages automatically generated from material catalogued by the participants of ARIS05
http://nickerson.icomos.org/aris05/

ASCix and ARIS05

ARIS05 was billed as an "International Course in Architectural Conservation, Heritage Recording, and Information Management" and was presented in Rome in the spring of 2005 by ICCROM with help from the Getty Conservation Institute.

ARIS05 computer lab

This was where the first trials of this software were undertaken and, as is often the case in early testing, conditions were encountered that were never contemplated during the design phase. Greatest of these was the multi user aspect of the course.

We had discussed splitting the database into front and back ends to it support a multi user environment but had put it aside, partly because of time constraints, but mostly because we felt it added a level of complexity we thought inappropriate. Surprisingly it worked pretty well in this situation (as long as the files are kept with their shadows) though it pointed out the need for some of the data integrity checks that had failed to make the prototype in time for the course.

Sixteen participants from sixteen different countries were given ASCix with instructions to build two data sets, one of the material they prepared during the heritage recording component of the course and another documenting their other activities while in Rome.

For the course component they were divided in to eight teams named pc1-8 (for the computer they were using) and for the course work they were asked to use only codes starting with their PC#. To combine the work of the different teams we simply copied their catalogued and shadow files into a common directory. This worked well except when shortcuts had been taken (things like rectifying an un-catalogued image and attaching it to a CAD drawing).

For the actual mechanism to keep the local catalogues in sync with the consolidated versions we resorted to the old stand-by, the batch file. This provided a one button solution to the problems of coordination and of doing backups.

However our attempts to coordinate the personal files was much less successful. In fact we failed though the attempt taught us a lot about the Data Integrity Tools we had hypothesized but had not yet implemented. In the next phase of development there will be easy ways to deal with missing or inconsistent Codes, Keywords and Authors and more comprehensive reporting of inconsistencies like invalid names or missing shadow files.

The problem was that, for their personal files, each participant had created a catalogue named for their user ID and the whole group was asked to collaborate in selecting codes and keywords under which to file the documentation of their visits to the touristic places in Rome. This didn't happen and our attempts to compile a composite catalogue of this material foundered on issues like inconsistent Codes and Keywords (Vatican / Vatikan for example),

There was also a lot of sloppy cataloguing. Apparently it was **too easy** to catalogue a file and a lot of second and third rate photos were catalogued, mostly without captions and often with wrong Codes. Forcing the inclusion of a caption would help as would making the user enter the Code each time instead of assuming, as is now the case, that the next file would probably belong to the same code. Then rejecting would become easier that cataloguing. We will probably make details like these configurable in later implementations.

Overall the experiment felt like a success though it did highlight many things that still need to be done. There are many adjustments, mostly minor, though there is the major task of moving from the prototype environment of MSAccess into a stand alone application that could be distributed to Heritage Recorders everywhere.

We are anxiously looking for feedback on the logic behind the program so far and would welcome partnerships to help take it to the next level.

Acknowledgements:

We wish to thank McGovern Heritage Archaeological Associates (MHAA) for their foresight in commissioning the original Field Data Management System from which this project grew (or shrank) and the ICCROM ARIS05 team and its leader, Ana Almagro in particular, for providing an excellent venue for the initial field trial by fire.

References:

Nickerson, S. 2005, ASCix Digital File Cataloguer
http://nickerson.icomos.org/ascix/

Nickerson, S. 1999a, A Database for the Long Term, *Presented to a Symposium on Byzantine Mosaics 27 March 1999 Krannert Art Museum, University of Illinois at Urbana-Champaign*
http://nickerson.icomos.org/euf/name.htm

Nickerson, S. 1999b, An Automatic Web Publishing Package for Complex Data Sets, *Proceedings of the CIPA XVII Symposium, Recife, Brazil*
http://nickerson.icomos.org/papers/cipa99.htm