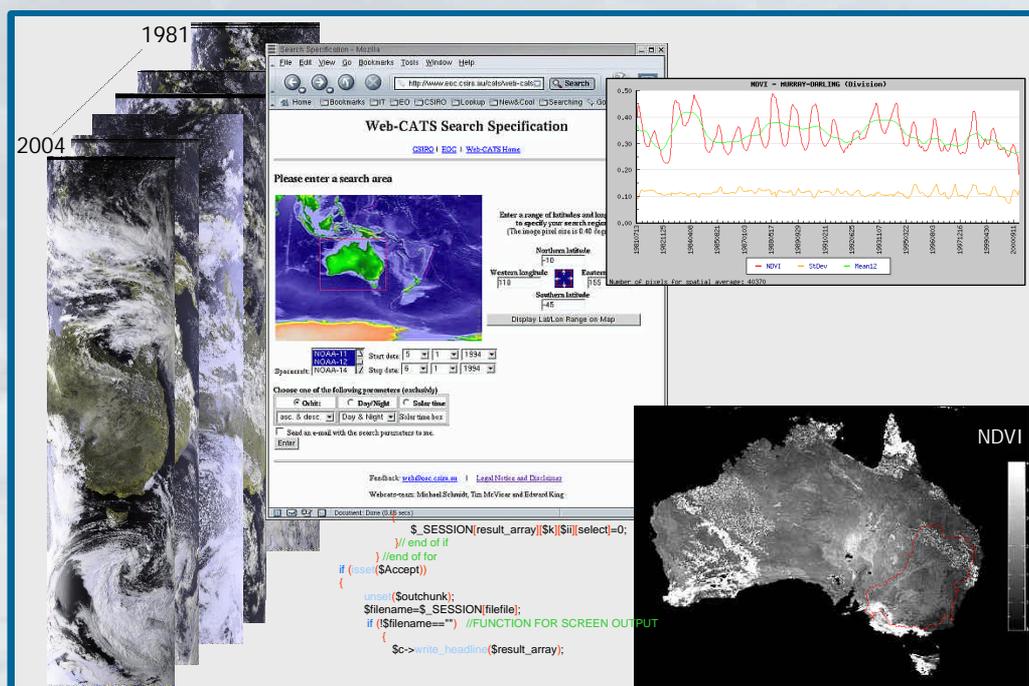


Development of a Web-based data and product delivery system for the CSIRO AVHRR Time Series (CATS): Architecture, Interface and Processing Descriptions

M. Schmidt, E.A. King and T.R. McVicar



Atmospheric Research

CSIRO Atmospheric Research Technical Paper No.66

Development of a Web-based data and product delivery
system for the CSIRO AVHRR Time Series (CATS):
Architecture, Interface and Processing Descriptions Tests

Michael Schmidt, Edward A. King and Tim R. McVicar

The National Library of Australia Cataloguing-in-Publication Entry

Schmidt, Michael.

Development of a web-based data and product delivery system
for the CSIRO AVHRR Time Series (CATS).

ISBN 0 643 06885 6.

1. Advanced very high resolution radiometers - Computer
network resources. 2. Earth sciences - Remote sensing. I.
King, Edward A. II. McVicar, Tim R. III. CSIRO
Atmospheric Research. IV. Title. (Series : CSIRO
Atmospheric Research technical paper (Online) ; 66).

550.285

Address and contact details: CSIRO Atmospheric Research
Private Bag No. 1, Aspendale Victoria 3195 Australia
Ph: (+61 3) 9239 4444; Fax: (+61 3) 9239 4400
Email: ar-enquiries@csiro.au

CSIRO Atmospheric Research Technical Papers may be issued out of
sequence. From July 2000, all new Technical Papers will appear on the
web site of CSIRO Atmospheric Research. Some Technical Papers will
also appear in paper form.

© CSIRO Australia Electronic edition, 2004

Abstract

Long time series of daily remote sensing data (i.e. 20 years and increasing) are useful for studies of intra- and interannual behaviour of surface components, such as vegetation cover and drought monitoring, and for the derivation of surface characteristics useful for climate modeling or land cover classifications. If a time series is well calibrated, data are especially valuable to detect and interpret long term environmental changes. These data can be used operationally as a decision support tool for policy makers. The Web-CATS (CSIRO AVHRR Time Series) project provides access to the Australian AVHRR (Advanced Very High Resolution Radiometer) data archive via the World Wide Web. The data archive consists of multiple satellite overpasses from several receiving stations in Australia for almost every day since July 1981 and is still growing. The data are held online, so that products can be processed on-demand in a user-customised manner with state-of-the-art algorithms. The user is presented with a list of the possible processing operations which are implemented in CAPS (Common AVHRR Processing Software). Users can search and browse a meta-database of the archived satellite passes and specify processing parameters for their desired data product. The architecture of Web-CATS is highly modular; so that the program (with slight modification) can be used to provide a Web-based delivery system for other long-term high-frequency environmental remote sensing sensors, such as MODIS (Moderate Resolution Image Spectrometer), ASTER (Advanced Spaceborne Thermal Emission and Reflection Radiometer) or future missions. Insight to this modular code is given in the appendices of this report, whilst an actual full electronic version of the package can be downloaded from the project website.

This report documents the existence of Web-CATS, which provides customised user access to the long time series of AVHRR data managed and corrected using best CSIRO's best practice algorithms. It provides a high-level technical overview of the functionality, including the underlying databases used by Web-CATS. This report does not aim to fill the role of a user-guide nor a detailed programming guide.

Contents

1	Introduction	1
2	Data and data organisation	3
2.1	Data	3
2.1.1	AVHRR data	3
2.1.2	The Australian AVHRR data	4
2.2	Database description	5
3	Setup and Installation	7
3.1	Hardware	7
3.2	Software	7
3.2.1	Apache	7
3.2.2	PHP	8
3.2.3	MySQL	8
3.2.4	Javascript	9
3.2.5	CAPS	9
4	WWW Interface	10
4.1	Introduction	10
4.2	Login and registration	12
4.3	Spatio-temporal data search	13
4.4	Product specification	16
4.5	Parameter verification	19
5	Data processing and delivery	21
5.1	Automatic product generation	21
5.2	Delivery	22

6	Product processing and delivery: An example of AVHRR GAC data	24
6.1	Australia wide AVHRR EOC-BPAL data description	24
6.2	Data application	26
7	Discussion and conclusions	27
	Bibliography	30
8	Appendix	34
8.1	A - Installation script	34
8.2	B - PHP script-example	35
8.3	C - Database tables	38

1 Introduction

Remotely sensed data for Earth Observation (EO) are now available for more than four decades (beginning with the successful CORONA missions in 1960 (Campbell, 1996)); the amount of information within these datasets is large and has increased tremendously due to an increasing number of satellite missions recently (Konecny, 2001; Petiteville et al., 2004). As the amount of information in every subject increases tremendously, improved and novel mechanisms to access and analyse the data are necessary. The increasing quantity of information demands dedicated solutions for storing and organising data; additionally as the data/information volumes increase, extra effort needs to be devoted to enabling the user community to access them (Zhou et al., 2001).

The rapid development in information technologies, and in particular communication technologies, has impacted, to various degrees, the geoscience environment (see Xue et al., 2002). The needs of data intensive geosciences (including remote sensing) are prime candidates for exploiting synergies arising from the momentum of these rapidly developing technologies (Xue et al., 2002). For large spatial datasets, the need for metadata storage and spatial data warehouses is especially obvious (Xue et al., 2002; Overton and Wragg, 2003).

The World Wide Web (WWW), originally a text-based communication platform, is developing towards a multimedia communication network (Kähkönen et al., 1999) and allows users to access a vast amount of information and data via easy-to-use web browsers. The gain in popularity of the WWW has led to it becoming a very important medium for research. It appears natural to use this medium to propagate and distribute data, products and applications in order to reach a wide group of users - non-experts as well as expert users - with specialisation in various disciplines (Carrara et al., 2003).

In geosciences, and remote sensing in particular, complex data integration processing steps are usually necessary before the data can be used in applications; for example with direct satellite broadcast and reception with real time data processing,

(geospatial-) database ingestion, calibration, navigation and geolocation of the data. Given a successful implementation of these basic steps, further product generation or refinement of the data can be performed (for further reading see Cracknell, 1997; King, 2003).

Previously users of remote sensing data were mostly provided with unprocessed data, which they then processed with their own algorithms or software packages, with only “in-house” defined standards. Consequently, no uniform product and data standard was achieved by different users, even though all worked with the same unprocessed data.

This report describes an approach towards a standardisation of data processing and product generation with state-of-the-art methods. The key in our approach is the dynamic product generation from base data on customer demand. Advances in information technologies permit addressing this issue as computer processing power and data storage capacities have increased. Coupled with advances in networking, image processing, process automation, and database management, the opportunities for information/data analysis and exchange have widened rapidly.

In this report we focus on the fully automatic operational satellite data processing and delivery of time series data of daily satellite images with a coverage of the Australian region, particularly AVHRR (Advanced Very High Resolution Radiometer) data recorded by the NOAA (National Oceanographic and Atmospheric Administration) polar orbiters and received at a number ground stations (King, 2003). The data are stored on-line and are processed on-demand with a computer cluster using state-of-the-art software modules of the Common AVHRR Processing Software (CAPS) package. This report is organised as follows: in chapter 2 the data and the corresponding relational database is outlined. Chapter 3 describes the software and hardware setup used in Web-CATS. The concept and functionality of the web interface is described in chapter 4, while chapter 5 gives an overview of the backend processing; chapter 6 gives an example of an application generated with the AVHRR time series, while discussions follow in chapter 7 where conclusions are also drawn. A major aim of this report is to allow readers to rebuild the frontend of Web-CATS for use within a similar environment and to encourage interest in applying the construct to other data searches. For this reason, all installation scripts, database tables and the source code of the web application are accessible.

2 Data and data organisation

2.1 Data

2.1.1 AVHRR data

The NOAA polar orbiters have carried AVHRR instruments operationally since the launch of NOAA-6 in 1979. The AVHRR sensor records data in 5 spectral bands of the electromagnetic spectrum: (1) Red 580 – 680 *nm*; (2) Near Infrared (NIR) 725–1100 *nm*; (3) 3.55–3.93 μm ; (4) 10.5–11.3 μm ; and (5) 11.5–12.5 μm . For further reading on the AVHRR instrument and NOAA series of satellites see Cracknell (1997). The spacecraft are in sun-synchronous polar orbits of $\approx 100\text{min}$ duration at an altitude of $\approx 700\text{km}$. Each orbit comprises an ascending and a descending component corresponding to whether the spacecraft is travelling northwards or southwards respectively.

The spatial resolution of High Resolution Picture Transmission (HRPT) AVHRR data is 1.1 km at the sub-satellite point, increasing to 5.4 km at the edge of the swath, at a scan angle of $\pm 55^\circ$ (see Cracknell, 1997, for further details). Global Area Coverage (GAC) data are recorded on-board with a nominal spatial resolution of 8 km . These data are the result of sub-sampling of the higher resolution HRPT data; a GAC pixel is the average of four HRPT AVHRR pixels, the next HRPT AVHRR pixel is skipped, then the next four HRPT AVHRR pixels are averaged to generate the second GAC pixel. This is repeated until the end of the line. The next two lines of HRPT AVHRR data are completely skipped and then the averaging process is repeated (NOAA, 1998). This means explicitly, that a GAC pixel is not an averaged value of the complete surface area observed with HRPT data. This is well illustrated by King (2003).

2.1.2 The Australian AVHRR data

The Australian HRPT data arise from archives recorded by receiving stations located at Alice Springs, Darwin, Hobart, Townsville and Perth. The dataset was originally compiled by the Australian Land Research Data Centre and continues to be maintained at the CSIRO Earth Observation Centre (EOC). Between 1981 and 1986 the data are daily GAC data (Cracknell, 1997) provided by NOAA. These data are also used to supplement the limited HRPT data from 1986 until 1992 when HRPT data began to be comprehensively archived. Since 1992, the HRPT data from the different stations have been combined by stitching the different segments from each station to eliminate redundancy and produce a single best-quality scene for each overpass (King, 2000). Only these continent spanning passes of stitched data are held in the online archive (King, 2003). The daily coverage is approximately a 50 million km^2 area, including the entire Australian land surface, New Zealand, East Timor, Papua New Guinea, most of Indonesia, parts of Malaysia and the Philippines and the surrounding ocean to at least 2000 km from the Australian Coast (see Figure 2.1). Since 1992, coverage of this area has been obtained four times daily since the data include both the day and night overpasses of satellites in both the morning and afternoon orbital planes (King, 2003).

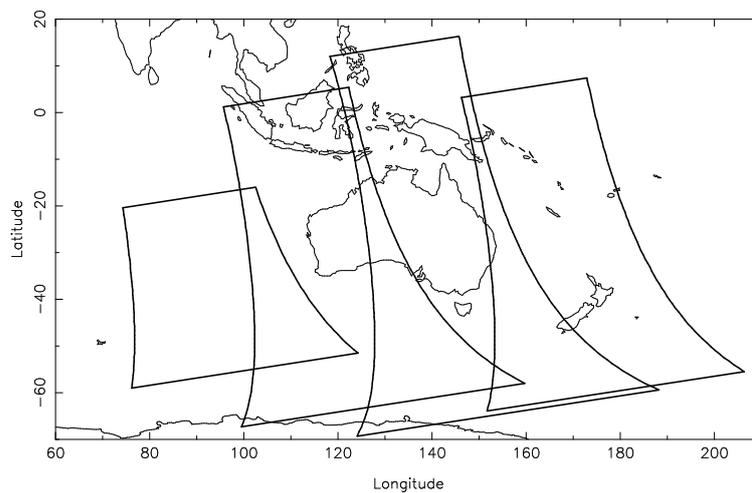


Figure 2.1: Typical daily coverage for one satellite in one direction, in this case for NOAA-14 ascending (afternoon) passes on 4 Jan 1996. The equivalent area is covered four times each day at approximately six hourly intervals by the two operational spacecraft both day and night.

During the more than two decades over which data have been acquired there have been numerous changes at the reception stations as a result of hardware replacement or operational changes. These are reflected in the variety of data formats, media formats and media types present in the raw archive at the EOC, see King (2003) for further details. At the time of writing, the volume of the stitched archive is 7 *Tbyte*, (approaching 100 000 scenes) and is stored on DLT (digital linear tape) in duplicate in ASDA (Australian Satellite Data Archive) format (Turner et al., 1998). When the data are ingested into the on-line storage the metadata for each overpass is automatically stored in a large relational meta-database, which is described below.

2.2 Database description

The large volume of AVHRR data requires a data warehousing concept to manage the data volume. A relational database holds the information and history of each scene and each single pass, including the time of image acquisition and the geographical coverage (described by an enclosing polygon computed from the satellite ephemeris data). The satellite data are distributed on the networked filesystem of the local disks of each computer within a cluster of personal computers running a Linux operating system.

The database entries can be accessed via SQL (Standard Query Language) queries (see Bowman et al., 1996); a web interface has been developed to instigate and update previous SQL queries (see chapter 4). The installation and configuration of the MySQL databases is described subsection 3.2.3.

The tables of the relational database used for the AVHRR data warehouse were designed to be comprehensive, time- and space-efficient. The database scheme has been normalised¹ to the greatest extent possible consistent with the need for extensibility and flexibility.

A brief description of the database is given here. A full listing of the tables is provided in Appendix C. The database is based around one large table (`avhrr_hrpt_scene`) which contains a single record with a unique identifier (primary key) for each scene in the archive. Each record contains detailed scene-specific information such as geographic extent and acquisition times. Pointers to other tables (foreign keys) de-

¹ Database normalisation is a series of steps followed to obtain a database design that allows for consistent storage and efficient access of data in a relational database. These steps reduce data redundancy and the chances of data becoming inconsistent. (see e.g. Bowman et al., 1996)

2 Data and data organisation

scribing characteristics common to many scenes (e.g. spacecraft, receiving station etc.) are also included. Several other tables recording file-locations and log-files etc. refer to the scene table.

3 Setup and Installation

3.1 Hardware

The web interface and database system runs on a SUNFire V100 blade server. The backend processing is currently taking place on a pilot cluster of AMD-Athlon-XP1600-based PCs running the Linux operating system. This cluster consists of 5 machines with a combined disk capacity of $1.75TB$, sufficient to hold the entire collection of afternoon-pass data. Plans are in train to expand this cluster to approximately 16 higher-performance machines with a disk capacity in excess of $8TB$ by the end of May 2004. This will provide storage for the entire Australian AVHRR dataset, together with significant processing capacity.

3.2 Software

No commercial software was used for Web-CATS; the concept is based on free, open source packages available over the internet. A prerequisite for a UNIX installation are the following packages: gcc, make, flex, bison, m4, autoconf, automake, Perl, gzip, tar and sed; most of which are included in standard installation or freely available from the internet (e.g. <http://www.sunfreeware.com/> - visited 17.02.2004).

3.2.1 Apache

Apache is an open source Web server. Apache runs on most UNIX-based operating systems, on UNIX/POSIX-derived systems, on AmigaOS, and on Windows. According to the Netcraft (<http://www.netcraft.com> - visited 14.02.2004) Web server survey in February, 2001, 60% of all Web sites were using Apache, making Apache more widely used than all other Web servers combined. For more details visit the Apache website (<http://www.apache.org> - visited 17.02.2004).

Apache version 1.2.27 was used in this application, Appendix A documents the installation options chosen.

3.2.2 PHP

PHP (Hypertext Preprocessor) is a server-side, HTML imbedded, cross-platform scripting language. It provides a way for a programmer to include instructions in HTML files to generate webpages dynamically. These instructions are read and parsed by a web-server (Apache in this case). The web-server interprets the PHP code and transforms it to HTML, which is then passed to the user's web-browser (Lerdorf, 2000). The PHP language is built upon concepts of other common languages, such as C and Perl. For further descriptions visit the PHP homepage (<http://www.php.net> - visited 17.02.2004).

For this application PHP was configured as a static module within Apache (see subsection 3.2.1). PHP contains a generalised database interface, in particular this enables full access to the MySQL database used in Web-CATS, permitting queries to be issued and results processed from within webpages.

In the current setup the (non standard) libraries GD, JpGraph, Juldat and Jpeg are required as the installation script shows (see Appendix A). The PHP version 4.3.2RC1 was used in this application. In Appendix B example PHP code is shown, with the complete and up to date set of code made available at the website <http://www.eoc.csiro.au/cats> by the end of April 2004.

3.2.3 MySQL

MySQL is a fast, multi-user, high performance, easy to use relational database system supporting SQL (DuBois, 1999). MySQL and PHP comprise, in combination, a powerful tool for dynamic web-page generation.

MySQL allows user and host management in the internal system tables, with which permissions for various purposes can be granted to single users for specific databases from given hosts. MySQL version 4.0.12 is used in this application. For further details see the MySQL website (<http://www.mysql.com> - visited 17.02.2004).

3.2.4 Javascript

Occasionally solutions were found to be more efficiently realised using Javascript, a scripting language from Netscape. JavaScript is intended to provide a quick and simple language for enhancing Web pages on the client side. JavaScript is embedded as a program in a web page that is interpreted and executed by the Web client (Stein, 1997). For more details see the Javascript webpage (<http://javascript.internet.com/> - visited 17.02.2004).

3.2.5 CAPS

CAPS was designed to implement standard processing algorithms for AVHRR data from the NOAA series polar orbiting satellites and developed as an extension to Tcl/Tk, and uses HDF and ASDA as the primary file formats (see Turner and Davies, 2000). Tcl is an embeddable scripting language which runs under a number of environments including UNIX and Windows. Tk is a GUI scripting language that is platform independent. The use of Tcl/Tk allows a rapid development cycle for CAPS by providing many necessary support functions and a simple method for writing GUI interfaces. CAPS performs six primary functions:

1. general array processing operations;
2. reading HRPT data;
3. reading and writing HDF files;
4. geographic location of AVHRR image pixels;
5. conversion of radiometer counts to radiance, reflectance and brightness temperature; and
6. mapping data from the satellite coordinate system to standard geographic coordinates.

See Turner et al. (1998) and Turner and Davies (2000) for further details or the CAPS web-site (http://www.dar.csiro.au/rs/avhrr_processing_software.htm - visited 17.02.2004).

4 WWW Interface

4.1 Introduction

The combination of HTML (HyperText Markup Language), PHP (Hypertext Pre-processor), MySQL relational database and Apache web server permits uncomplicated access to the satellite overpass metadata (stored in the database) with any web browser. Database requests can be submitted from a remote computer, by an authorised user, through a web interface generated with PHP and HTML. The result of the user request is handled by PHP and sent back in HTML to the client computer for display; a schematic diagram of the process is shown in Figure 4.1.

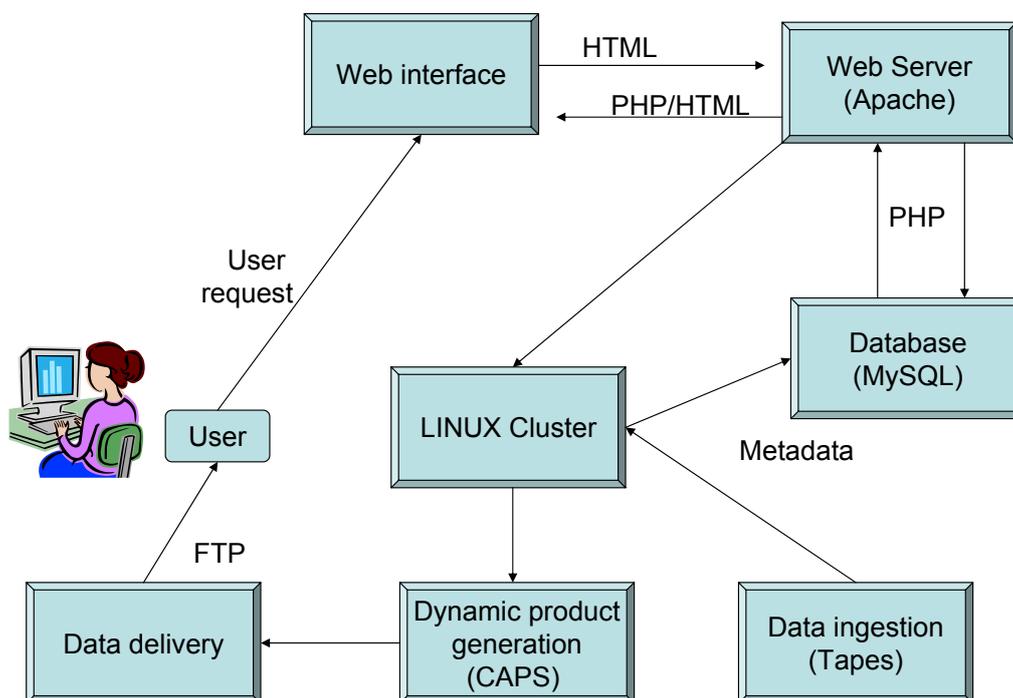


Figure 4.1: Software and hardware architecture of Web-CATS to allow user defined, customised data and product generation.

The web-functionality was achieved by utilising object oriented programming facilities within PHP. Grey boxes in Figure 4.2 represent PHP scripts (filenames ending with .eoc suffix), the blue containers represent the MySQL databases (AVHRR0002 and Webcats), the flag symbols represent object classes. The white box labelled session_functions.inc represents a library of PHP functions. All program parts are linked relatively with only one main configuration file (white box labelled parameters.inc) that needs updating if the pages move to another location or are used for another satellite data set. Figure 4.2 provides an overview of the scripts developed for Web-CATS.

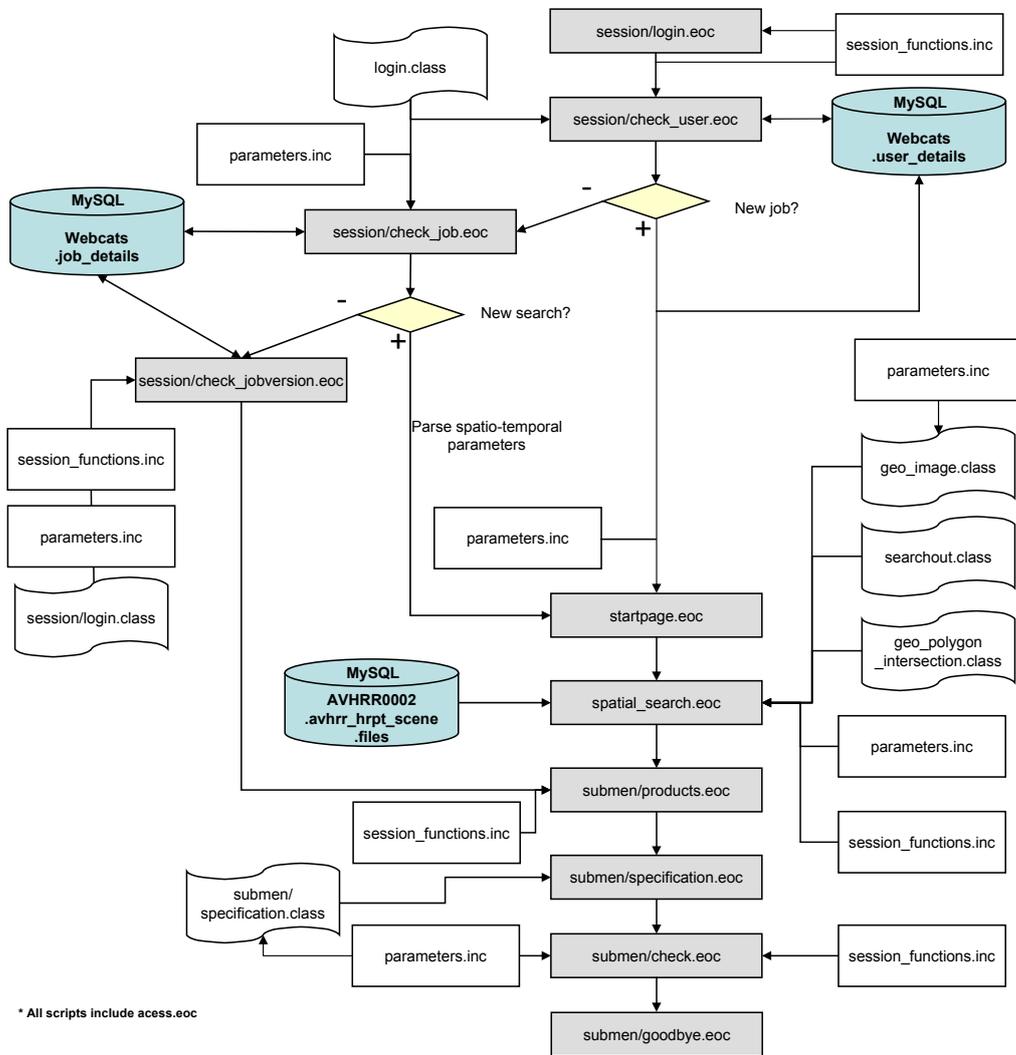


Figure 4.2: Flow chart of the interacting webpages (description in the text).

Help pages guide a user through this web application, explain every step in detail and give necessary background information of the satellite system and remote sensing in general.

4.2 Login and registration

Users are required to register before accessing the system. Registration enables individual users to be identified, and for user-specific data to be stored for use during subsequent visits. Each time a user connects to the web interface, they identify themselves by means of a username and password (`login.eoc` in Figure 4.2). As they then proceed through a sequence of web pages choosing their data and processing options, they are tracked by means of a session cookie stored in their browser. When initiating a data search and processing request the user may provide a name for the request which is stored with the request parameters in the database. This permits re-use of the parameter specification in subsequent searches (`check_job.eoc` in Figure 4.2); for example enabling the same products to be regenerated after more data becomes available, or using the same processing parameters to produce a similar product at a number of different locations or times.

4.3 Spatio-temporal data search

After registration, the user is directly guided to the product selection page (`products.eoc` in Figure 4.2) or asked to specify parameters for a new spatio-temporal search (`startpage.eoc` in Figure 4.2). In the spatio-temporal search page, shown in Figure 4.3, the user is asked to define their geographic region of interest (ROI) and their temporal search window. On the same page the user has the option to either search for all available data (default) or to specify which particular satellites (currently: NOAA-6, 7, 8, 9, 10, 11, 12, 14, 15, 16, 17), overpass direction (ascending or descending orbit) or time of day should be included. The internal query searches first for matching single passes in one SQL query for time, satellite and overpass. Then the geographical search routines check each single pass of the result for overlap with the user-specified ROI. The geographical search was programmed within PHP and is thus not dependent on supplementary software packages including a (potentially slow) geographical information system for polygon intersection

testing. All bounding polygons for the scenes are stored within the database (see table `avhrr_hrpt_scene` in Appendix C for details). The search algorithm tests for intersection of the line segments from the bounding polygon of each scene with the those from the ROI. A test is also performed to check whether one polygon completely encloses the other (without boundary intersection). The algorithm stops as soon as a positive intersection is found; whereupon the next match from the SQL query is tested.

Web-CATS Search Specification

[CSIRO](#) | [EOC](#) | [Web-CATS Home](#)

Please enter a search area

Enter a range of latitudes and longitudes to specify your search region.
(The image pixel size is 0.40 degrees)

Northern latitude:

Western longitude: Eastern longitude:

Southern latitude:

Spacecraft: NOAA-11 NOAA-12 NOAA-14

Start date:

Stop date:

Choose one of the following parameters (exclusively)

Orbit: Day/Night: Solar time

Send an e-mail with the search parameters to me.

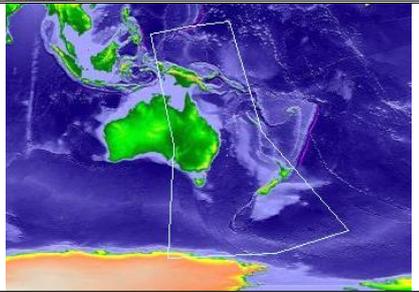
Feedback: web@eoc.csiro.au | [Legal Notice and Disclaimer](#)

Webcats-team: Michael Schmidt, Tim McVicar and Edward King

Figure 4.3: Web-interface of the spatio-temporal AVHRR data search.

The stitched overpasses that pass all tests (spatial, temporal and satellite overpass), are displayed in a table with a list of relevant metadata including a quicklook image and footprint. Table 4.1 shows an example of a search output (spatial_search.eoc in Figure 4.2).

Table 4.1: Example of a single search result of a database query.

Quicklook	
Footprint	
processing_type	stitched level0-browse
Spacecraft	NOAA-11
Begin Date (GMT)	1994 1 5 6h 15min 27sec
End Date (GMT)	1994 1 5 6h 38min 13sec
Orbit Nr.	27220
centre_dir	A
daynight	D
nwlat	9.3363854
nwlon	130.3785672
nnlat	11.9146278
nnlon	144.1695848
nelat	13.7732362
nelon	157.8526883
celat	-24.6735171
celon	168.4892052
selat	-58.7260578
selon	-161.7396998
snlat	-66.6650142
snlon	173.5305985
swlat	-68.2883989
swlon	136.6041628
cwlats	-29.4949681
cwlon	138.1115583

The SQL queries used for the spatio-temporal part of this application use only the tables `avhrr_hrpt_scene` and `file` (see Appendix C for more details). Figure 4.4 shows an example of a data search for January 05 - 06, 1994 (Julian Days 2449357.5 - 2449359.5) for all satellites and overpasses, corresponding to Table 4.1.

Select

```
file.location, file.name, file.processing_type, avhrr_hrpt_scene.sat_id, avhrr_hrpt_scene.start_jd,
avhrr_hrpt_scene.end_jd, avhrr_hrpt_scene.start_orbit, avhrr_hrpt_scene.centre_dir,
avhrr_hrpt_scene.daynight, avhrr_hrpt_scene.nwlat, avhrr_hrpt_scene.nwlon,
avhrr_hrpt_scene.nnlat, avhrr_hrpt_scene.nnlon, avhrr_hrpt_scene.nelat, avhrr_hrpt_scene.nelon,
avhrr_hrpt_scene.celat, avhrr_hrpt_scene.celon, avhrr_hrpt_scene.selat, avhrr_hrpt_scene.selon,
avhrr_hrpt_scene.snlat, avhrr_hrpt_scene.snlon, avhrr_hrpt_scene.swlat, avhrr_hrpt_scene.swlon,
avhrr_hrpt_scene.cwlat, avhrr_hrpt_scene.cwlon, avhrr_hrpt_scene.nwlat, avhrr_hrpt_scene.nwlon
from avhrr_hrpt_scene, file
where
avhrr_hrpt_scene.start_jd>='2449357.5' and avhrr_hrpt_scene.end_jd<= '2449359.5' and
file.scene_id=avhrr_hrpt_scene.scene_id and avhrr_hrpt_scene.flags like '%0%' and
file.processing_type like '%-browse' order by avhrr_hrpt_scene.start_jd
```

Figure 4.4: Example of a SQL query generated from user input request for a data search from 0000 GMT January 05 1994 to 2400 GMT January 06, 1994.

The user is presented with the option of inspecting the search results and interactively excluding single passes from the data to be used for subsequent product generation. The footprint-images are generated as a raster image overlaid with vector data of the spatial extent of the satellite coverage (currently a polygon consisting of 8 points) and passed to the users' browser.

4.4 Product specification

Having selected a set of data to process, the user must also specify the processing required in order to generate the desired product(s), which may be a particular geophysical parameter or thematic map (products.eoc in Figure 4.2). A range of different products and processing strategies have been implemented to enable as wide a choice as possible. As an example, a user requesting NDVI (Normalised Difference Vegetation Index) data (see Tucker, 1979) could choose satellite overpasses at a single epoch in satellite projection with atmospheric correction, or alternatively, atmospherically and geometrically corrected data in UTM projection composited using the maximum value algorithm (see Cracknell, 1997, for further reading) over

10-day intervals. The Web-CATS interface and processing system is flexible enough to meet both these, and a range of other user-customised requirements.

Although for many products several options are implemented, in order to allow non-expert users to obtain a standard product, a default option is provided with a set of recommended processing parameters chosen *a priori*. In addition to the default values, links to online help pages and other background information and documentation are provided. Products derived from the AVHRR data are divided into direct products, as derived from the single bands, and synthesised products, which are a combination of single channels and/or processed data, possibly including auxiliary data (as in the case of the atmospheric corrected datasets). Table 4.2 shows the availability of direct data products for each raw data channel.

Table 4.2: Direct data products per spectral channel as directly derived from AVHRR raw data.

Product	ch 1	ch 2	ch 3a	ch 3/3b	ch 4	ch 5
Digital number	×	×	×	×	×	×
Radiance	×	×	×	×	×	×
Brightness temperature				×	×	×
Reflectance	×	×	×			
Normalised reflectance	×	×	×			

Table 4.3: List of direct products.

Product	Status of realisation
Digital number	implemented
Radiance	implemented
Brightness temperature	implemented
Reflectance	implemented
Normalised reflectance	implemented
Satellite view angles	implemented
Solar illumination angles	implemented
Recording time (per pixel)	proposed

Table 4.3 summarises the implementation status of the direct products (at the time of writing). The solar illumination and satellite view angles are computed from satellite ephemeris data. In addition, the acquisition time of each pixel will be available.

For many of the products (both direct and synthesised) there are often a number of choices for algorithms and ancillary data to be used in the processing; in order to view the options visit the Web-CATS website, as these options are being extended as new methods become available.

Table 4.4: List of synthesised products (see text for explanation).

Product	Status of realisation
Vegetation indices (NDVI, etc.)	implemented
Other Indices (NDSI, etc.)	implemented
Albedo	implemented
Surface temperature	implemented
BRDF corrected reflectance	implemented
Cloud mask	implemented
Atmospheric correction	end of March 2004

Table 4.4 shows the synthesised products implemented and planned; where BRDF stands for bidirectional reflectance distribution function, see Roujean et al. (1992) for more details. NDSI represents the normalised differenced snow index, see Romanov et al. (2000) for more details. General options for all products are projection, output pixel-size ($> 500m$), data resampling method (nearest neighbour or bilinear interpolation) or output file format (HDF or GeoTiff [proposed]). For time series processing the option of temporal compositing is provided. The user has the option to choose between different compositing methods (Table 4.5) and compositing intervals (Table 4.6). Maximum value compositing is explained in Figure 6.1, minimum and mean value composites are calculated analogously. The MVZAC stands for minimum view zenith angle composite see Lovell et al. (2003) for full details.

Table 4.5: List of compositing methods (see text for explanation).

Method	Status of realisation
Maximum Value Composite	implemented
Minimum Value Composite	implemented
Mean Value Composite	implemented
MVZAC	implemented

Table 4.6: List of compositing intervals.

Product	Status of realisation
1..n days ($n < 100$)	implemented
Three per month ($\cong 10$ day)	implemented
Bi-monthly ($\cong 15$ day)	implemented
Monthly	implemented
2-Month	implemented
3-Month *	implemented

* for small search regions only

4.5 Parameter verification

The final set of product generation parameters is shown in a check-out page with an option to either accept these parameters (and proceed with the product generation), or to alter the setup. The final list of input files is saved on the server and recorded in the Web-CATS database together with the search and product generation parameters and the user-provided job name. An example of a parameter file is given in Figure 4.5.

USERNAME: SCH438
JOBNAME: MICHAEL_1
VERSION: 000001
ORBIT: 0
SATELLITE: ALL
START_DAY: 1
START_MONTH: 1
START_YEAR: 1994
STOP_DAY: 1
STOP_MONTH: 1
STOP_YEAR: 1994
START_JD*: 2449353.5
END_JD*: 2449354.5
SEARCH_POLYGON: 20.000,80,20.000,220,-80.000,220,-80.000,80
PRODUCT: NDVI
PROJECTION: UTM
PIXELSIZE: 1.100
RESAMPLING: NN
BRDF_TYPE: SIMPLE
ATCOR_TYPE: MITCHELL
COMP_TYPE: MAX
WHICH_COMP: CONSTANT
COMPOSITING_INTERVAL: 10
CLOUD_TYPE: MASK

Figure 4.5: Example of a parameter-file generated by Web-CATS due to user specifications (*JD=Julian day). The explanatory text-header is not shown here.

5 Data processing and delivery

5.1 Automatic product generation

The processing system, CAPS, was developed by CSIRO to promote the production of standard AVHRR data products for the Australian continent by incorporating best practice methods (Turner et al., 1998; Turner and Davies, 2000). CAPS can import ASDA format data and is operated within the EOC on a cluster of networked PCs running the LINUX operating system. Within CAPS, processing routines are called in a command line or script mode, which allows the list of data files, with the corresponding set of processing parameters, to be sent to a script generating a CAPS processing file. The processing job is then queued into a list of jobs, that are processed in parallel by the LINUX cluster (see King et al., 2004, for further details). The hardware was described in section 3.1. An example of a CAPS script is shown in Figure 5.1.

```

priority          9
job_id           qld_ndvi
reference_date    19950126
interval         9
start_date       20030815
end_date         20031130
outdisk          chief:/data/ch1
data_path        alsp/16a
xystep           0.01
min_lat          -30.0
max_lat          -10.0
min_lon          141.0
max_lon          155.0
[compos]
composite_script /pool/tasks/caps/scripts/compos.tcl
composite_type   NDVImvc
input_file_pattern ah*.hdf
lat_chunk_step   10.0
lon_chunk_step   14.0
[caps]
chunk_lines      5000
angle_step       1
re_outputs       {(warpedInterpolated_(sun|satellite)(Zenith|Azimuth)Grid$|
warpedNearest_normalisedReflectance_avhrr_(1|2)$|warpedNearest_cloud_mask$)}
modules          getAVHRR geonav calSol calTIR bt expandGrid isLand cloudAVHRR warp putHDF

```

Figure 5.1: Example of a CAPS script automatically generated by Web-CATS.

5.2 Delivery

When the product generation is complete an e-mail notification is sent to the user. The data can be accessed via file transfer protocol (FTP) and remains on the web-server for 7 days before removal. A file containing the data selection and processing parameters accompanies the data products. The data are exported in HDF file formats, which can be imported into the common data processing software packages or tools. Figure 5.2 shows an example of an user required data product, generated by CAPS.

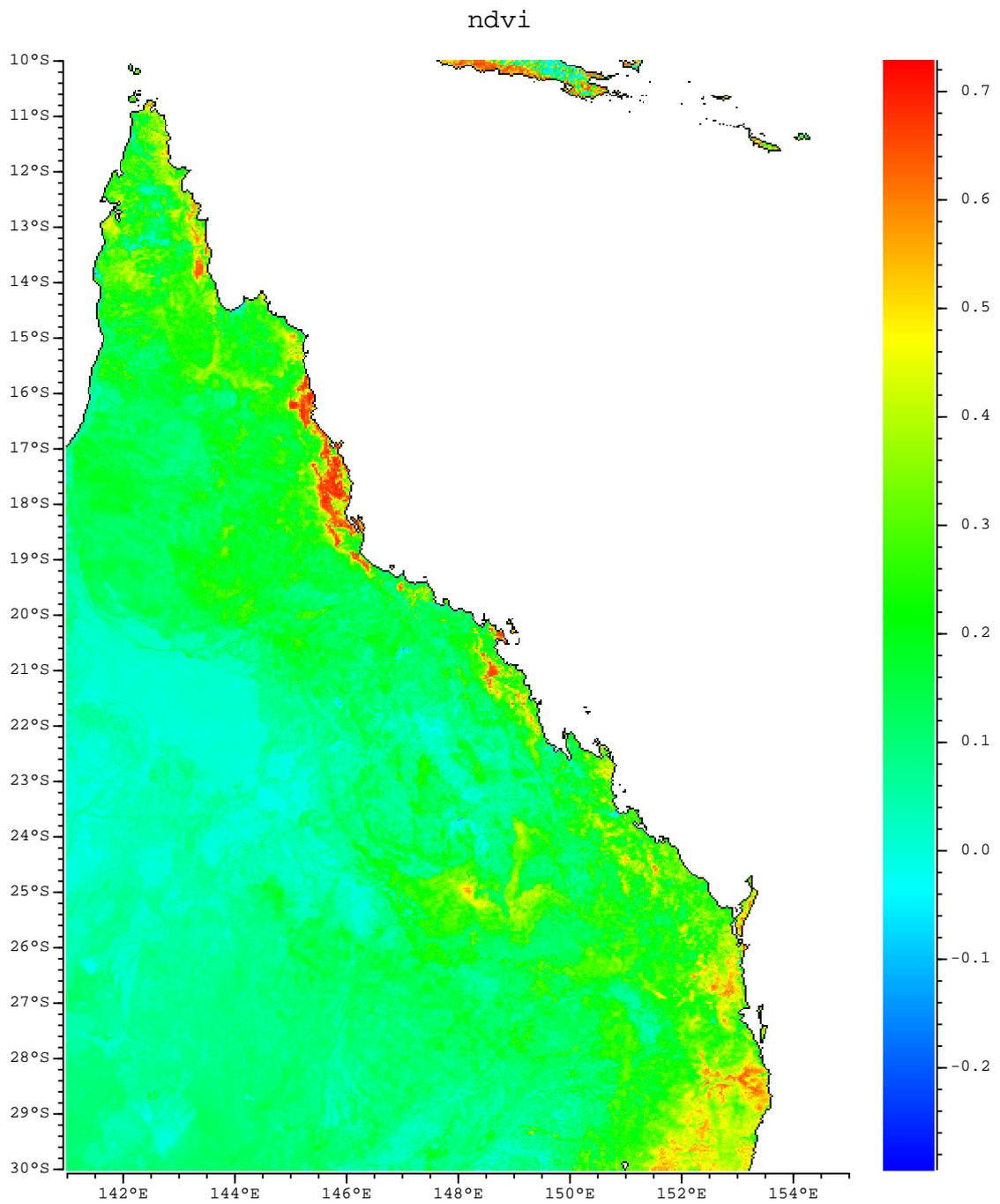


Figure 5.2: Example of an user specified data product generated with CAPS: a NOAA-16 MVC NDVI image of Queensland over the period from November 13 to November 21, 2003.

6 Product processing and delivery: An example of AVHRR GAC data

In this chapter an example of a data service is described. This pilot system allows near real-time time series data analysis and is publicly accessible by the following URL:

http://www.eoc.csiro.au/cats/bpal/AVHRR_GAC/index.eoc

6.1 Australia wide AVHRR EOC-BPAL data description

The Earth Observation System (EOS) Pathfinder Program of NASA (National Aeronautics and Space Administration), in cooperation with NOAA, at NASA's Goddard Space Flight Center, has processed 12 years of GAC data (Agbu and James, 1994; James and Kalluri, 1994). The Pathfinder AVHRR Land (PAL) GAC data starts in July 1981 and extends until September 1994.

For Australia the time series of the PAL data has been extended from September 1994 to present by generating GAC-like data from the NOAA-14 and NOAA-16 afternoon passes. The compositing interval is 9 days and the stitched HRPT data (King, 2000) were resampled with a nearest neighbor algorithm to match the *8km* GAC spatial resolution (Lovell et al., 2003).

To minimise the effects of clouds and atmospheric contaminants for each pixel, composite images using the maximum NDVI value over an interval was used (Holben, 1986), see Figure 6.1. Additionally, only data within 42° of nadir are used in the composite to minimise spatial distortion and view and illumination geometry effects at the edge of a scan. Remaining cloud artifacts were reduced by filtering the 1981 - present time series data (both PAL and the CAPS-processed data) using a modified Best Index Slope Extraction (BISE) method (Viovy et al., 1992), for further details

see Lovell and Graetz (2001). The resulting dataset is referred to as BPAL.

Day 1		Day 2		Day 3		Day 4		Result	
0.08	0.22	0.05	0.24	0.07	0.23	0.06	0.23	0.08	0.24
0.23	0.19	0.27	0.07	0.06	0.06	0.31	0.11	0.31	0.19

Figure 6.1: Schematic of the MVC (Maximum Value Compositing) procedure for a 4-day composite. The numbers in the first four (left to right) 2×2 arrays represent single overpass NDVI values at the same locations for four days. The result of a maximum NDVI compositing procedure for the 4 day observation period is shown in the final 2×2 array: the top-left pixel is the maximum of the 4 input top-left pixels, the top-right pixel is the maximum of the 4 input top-right pixels, and so on.

Polygons from States and hydrological catchments on different scales were obtained from the National Land and Water Resource Audit (<http://www.nlwra.gov.au> - visited 23.02.2004). Regional averages of NDVI data were calculated for every catchment and every composite. Catchments smaller than $100km^2$ were excluded.

6.2 Data application

To obtain NDVI time series statistics, a user can set a time span to be displayed and choose between pre-defined areas based on the 8 state boundaries or on hydrological catchments on differing scales: Divisions (12), Regions (77) and Basins (245). Figure 6.2 shows an example of the resulting figure (including the selected options of the mean NDVI, the standard deviation and the running mean of 12 months).

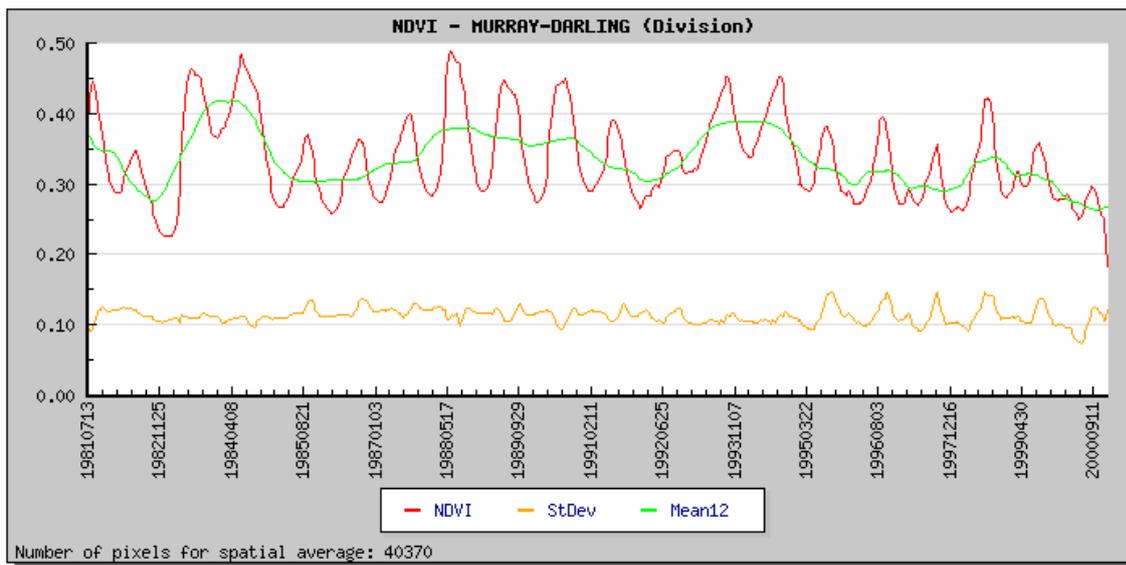


Figure 6.2: Screen capture from the averaged NDVI time series spatially averaged over the Murray-Darling Basin. Displayed is the mean NDVI, the standard deviation and the running mean of 12 months.

The NVDI spatial averages are pre-calculated and stored as ASCII files on the web-server. The raw data, used in the generation of this graph, can be downloaded as text or MS EXCEL files, for the generation of graphs according to each user's wishes. This tool provides a quick insight of the temporal behaviour of the spatial averaged NDVI data as an environmental indicator e.g. for droughts (McVicar et al., 2003) and is useful for decision making processes. In combination with satellite direct broadcast and dynamic updates, data analysis in near-real time becomes possible.

7 Discussion and conclusions

Due to recent (and ongoing) developments in information technology, the creation of a WWW-based data delivery service for large data sets is now readily achievable. The huge advances in web technologies and the commoditisation of web application environments permits the development of sophisticated interfaces that greatly ease user-interaction with data sets. The more advanced web scripting languages (such as PHP) enable flexible and modular development of these interfaces and greatly facilitate rapid extension to incorporate new functionality.

For the first time in Australia, AVHRR data from different Australian receiving stations are brought together. The data were substantially improved in value by stitching the whole archive to generate the best possible data coverage in time and space (King, 2000). Data consistency, dynamic product generation, with state-of-the-art methods as implemented in CAPS, can now be applied to this outstanding dataset. The (pre-) processing is, from a users point of view, outsourced so that no additional investment in either software, hardware or human remote sensing experts is necessary. These three components usually contribute substantially to the cost of projects.

This “outsourcing” (and centralisation) of the basic processing, has several other advantages:

- Users need not be concerned with storage media or data format issues;
- Any lack of calibration or coefficient files, which would normally hamper the product generation and cause project delays, are dealt with once for all users;
- Updating of these files (involving download and installation), activities that are at best tangential to data analysis, are no longer of concern to users;
- State-of-the-art methods are implemented in CAPS as standard routines;
- Time series data over a more than 20 year period from the same satellite mission are held in one system; and

-
- Every dataset is produced with a common set of methods and can be re-produced identically because of the user management and associated file and processing records held in the user database.

A potential draw-back from the user point of view might be the dependence on CSIRO to provide this service. This can however be seen as an opportunity to interact and collaborate with CSIRO as a scientific agency and request implementation of routines for specific data products. By contributing anomaly reports and new algorithms for incorporation into the system, users can participate in improvement and evolution of the system to meet a wider variety of needs for a larger group of potential users. Even so, as new functionality is incorporated into CAPS it becomes available for users who wish to do their own processing: Executable versions of CAPS are freely available and can be downloaded (see subsection 3.2.5) by any user.

Web-CATS caters for expert users by providing rich functionality and a wide range of processing choices. However, the needs of non-expert users who simply require a standardised product are also met through the provision of standard default parameters for a number of basic products. Regardless of the level of knowledge, help buttons and documentation pages throughout the data selection process are provided to assist the user.

A major benefit of Web-CATS is that the longest existing time series of satellite data of the Australian continent can be accessed in near real-time by relatively inexperienced remote sensors. Information derived from data products, such as drought maps or fire maps can be used to directly influence decision makers. This valuable record of daily satellite data for more than 20 years, in combination with web access, opens many avenues in several environmental fields for further application and research development (McVicar and Jupp, 2002; McVicar et al., 2003; Lu et al., 2003). Thus, the list of products implemented in the Web-CATS is under constant update as improvements and new product routines are implemented in CAPS. The webpages and database architecture are largely transferable to other EO satellite data such as MODIS (Moderate Resolution Image Spectrometer) or the forthcoming NPOESS (National Polar-orbiting Operational Environmental Satellite System) data (Townshend and Justice, 2002). This service is located under the following address: <http://www.eoc.csiro.au/cats> .

Acknowledgements

We would like to kindly thank all those who have helped, supported and encouraged us to do this work. The CSIRO AVHRR time series, and the best practice tools to process that data is the result of many years work by past and current colleagues within CSIRO, including:

Ian Barton, Denis O'Brien, Susan Campbell, Steven Clift, Harvey Davies, Mac Dille, Mary Edwards, Dean Graetz, Ian Grant, David Jupp, Jeff Kingwell, Jenny Lovell, Ross Mitchell, Alan Pearce, Fred Prata, Chris Rathbone, Michael Raupach, Paul Tildesley, Peter Turner and Murray Wilson.

With thank them all (and any we have forgotten who worked on the data prior to our involvement with it) for the valuable part each has played in necessary *a priori* research that Web-CATS is based upon.

Bibliography

- Agbu, P. A. and James, M. E. (1994). *The NOAA / NASA Pathfinder AVHRR Land Data Set Users Manual*, Goddard Distributed Active Archive Centre. NASA, Goddard Space Flight Centre, Greenbelt, USA.
- Bowman, J., Emerson, S. L., and Darnovsky, M. (1996). *The practical SQL Handbook*. Addison-Wesley, 3 edition.
- Campbell, J. B. (1996). *Introduction to remote sensing*. Taylor and Francis Ltd., London.
- Carrara, P., Fresta, G., and Rampini, A. (2003). Banco: an svg-based approach to create web sites for the management of remote sensing, spatial and non-spatial data. *International Journal of Remote Sensing*, 24(20):3903–3916.
- Cracknell, A. P. (1997). *The Advanced Very High Resolution Radiometer*. Taylor and Francis, London.
- DuBois, P. (1999). *MySQL*. New Riders Publishing, Indianapolis, USA.
- Holben, B. (1986). Characteristics of maximum-value composite images from temporal AVHRR data. *International Journal of Remote Sensing*, 7:1414–1434.
- James, M. E. and Kalluri, S. N. V. (1994). The pathfinder AVHRR land data set: an improved coarse resolution data set for terrestrial monitoring. *International Journal of Remote Sensing*, 15:3347–3363.
- Kähkönen, J., Lehto, L., Kilpeläinen, T., and Sarjakoski, T. (1999). Interactive visualisation of geographical objects on the internet. *International Journal of Geographical Information Science*, 13:429–438.

- King, E. A. (2000). Sticking the Australian 1-km archive. In McVicar, T. R., editor, *Proceedings of the 10th Australasian Remote Sensing and Photogrammetry Conference, 25 August 2000, Adelaide, Australia*, pages 41–50. CSIRO, Canberra, Australia.
- King, E. A. (2003). *The Australian AVHRR Data Set at CSIRO/EOC: Origins, Processes, Holdings and Prospects*. CSIRO CAR/EOC Report 2003/04, Canberra, Australia. 48 p http://www.eoc.csiro.au/tech_reps/2003/tr2003_04.pdf.
- King, E. A., Schmidt, M., and McVicar, T. R. (2004). Self balancing distributed processing system for remote sensing data. in preparation.
- Konecny, G. (2001). High resolution mapping from space. In *High resolution mapping from space*, E-proceedings of ISPRS workshop. Hannover, Germany.
- Lerdorf, R. (2000). *PHP Pocket Reference*. O’ Reilly, Beijing, Cambridge, Farnham, Köln, Paris.
- Lovell, J. L. and Graetz, R. D. (2001). Filtering pathfinder AVHRR land NDVI data for Australia. *International Journal of Remote Sensing*, 22:2649–2654.
- Lovell, J. L., Graetz, R. D., and King, E. A. (2003). Compositing AVHRR data for the Australian continent: Seeking best practice. *Canadian Journal of Remote Sensing*, 29:770–782.
- Lu, H., Raupach, M. R., McVicar, T. R., and Barrett, D. J. (2003). Decomposition of vegetation cover into woody and herbaceous components using AVHRR NDVI time series. *Remote Sensing of Environment*, 86:1–18.
- McVicar, T. R., Briggs, P. R., King, E. A., and Raupach, M. R. (2003). *A Review of Predictive Modelling from a Natural Resource Management Perspective: The Role of Remote Sensing of the Terrestrial Environment*. CSIRO Land and Water Client Report to the Bureau of Rural Sciences, Canberra, Australia. 26 p http://www.clw.csiro.au/publications/consultancy/2003/-Review_Of_Remote_Sensing.pdf.
- McVicar, T. R. and Jupp, D. L. B. (2002). Using covariates to spatially interpolate moisture availability in the Murray-Darling Basin: a novel use of remotely sensed data. *Remote Sensing of Environment*, 79:199–212.

- NOAA (1998). *NOAA Polar Orbiter Data User's Guide*. Internet: <http://www2.ncdc.noaa.gov/docs/podug/>, (15.01.2004).
- Overton, I. and Wragg, B. (2003). Web-based searching and cataloguing for map collections and the map trade industry. In *Proceedings of Spatial Science Conference: Spatial Knowledge without Boundaries, 22-26 September 2003*. CSIRO, Canberra, Australia.
- Petiteville, I., Caspar, C., Dessard, V., and Barois, O. (2004). Web mapping techniques applied to the visualization of ESA satellite products. *Acta Astronautica*, 54:177–181.
- Romanov, P., Gutman, G., and Csiszar, I. (2000). Automated monitoring of snow cover over North America using multispectral satellite data. *Journal of Applied Meteorology*, 39:1866–1880.
- Roujean, J.-L., Leroy, M., and Deschamps, P.-Y. (1992). A bidirectional reflectance model of the earth's surface for the correction of remote sensing data. *Journal of Geophysical Research*, 97:20,455 – 20,468.
- Stein, L. D. (1997). *How to set up and maintain a Web Site*. Addison Wesley, Massachusetts, USA, 2nd edition.
- Townshend, J. R. G. and Justice, C. O. (2002). Towards operational monitoring of terrestrial systems by moderate-resolution remote sensing. *Remote Sensing of Environment*, 83:351–359.
- Tucker, C. J. (1979). Red and photographic infrared linear combinations for monitoring vegetation. *Remote Sensing of Environment*, 8:127–150.
- Turner, P. J. and Davies, H. L. (2000). Advances in CAPS. In McVicar, T. R., editor, *Proceedings of the 10th Australasian Remote Sensing and Photogrammetry Conference, 25th August, Adelaide, Australia*, pages 71–80. CSIRO, Canberra.
- Turner, P. J., Davies, H. L., Tildesley, P. C., and Rathbone, C. E. (1998). Common AVHRR processing software (CAPS). In McVicar, T. R., editor, *Proceedings of the Land AVHRR Workshop, 9th Australasian Remote Sensing and Photogrammetry Conference, 24th July, Sydney, Australia*, pages 51–58. CSIRO, Canberra.

- Viovy, N., Arino, O., and Belward, A. S. (1992). The Best Index Slope Extraction (BISE): A method for reducing noise in NDVI time-series. *International Journal of Remote Sensing*, 13:1585–1590.
- Xue, Y., Cracknell, A., and Guo, H. (2002). Telegeoprocessing: the integration of remote sensing, Geographic Information System (GIS), Global Positioning System (GPS) and telecommunication. *International Journal of Remote Sensing*, 23:1851–1893.
- Zhou, X., Yates, J., and Chen, G. (2001). Using visual spatial search interface for www applications. *Information Systems*, 26:61–74.

8 Appendix

8.1 A - Installation script

In this appendix the configuration and installation script of Apache with PHP as static module is described. Apache/PHP was configured to execute files ending with the .eoc suffix due to the line: [AddType application/x-httpd-php .eoc] in the Apache configuration-file httpd.conf as described below.

```
# last modified: 08.07.2003
#download apache, php and extract in desired directory!

cd ../php_DIR

./configure --with-mysql --with-apache=../apache_DIR --enable-calendar --enable-bcmath --enable-ftp --enable-mbstring --
enable-mbregex --with-jpeg-dir=/lib_DIR/ --with-tiff-dir=/lib_DIR/ --with-gd --with-zlib --with-regex=system

make

make install

cd ../apache_DIR/

./configure --prefix=www_DIR --activate-module=src/modules/php4/libphp4.a --with-perl=perl_DIR

make
make install

cd ../php_DIR
cp php.ini-dist /lib_DIR/php.ini

#modify the file /lib_DIR/php.ini to set
register_globals = on

#edit httpd.conf in the folder /www_DIR/conf/ and add the line
AddType application/x-httpd-php .php

#and the line
AddType application/x-httpd-php .eoc

#change the line:
DocumentRoot "/www_DIR/htdocs" to the desired www folder e.g. /www
```

Figure 8.1: Configuration and installation of Apache with PHP as static module.

8.2 B - PHP script-example

This appendix shows an example script of the PHP code developed for this application. The complete code will be available at <http://www.eoc.csiro.au/cats> by the end of April 2004.

```

check_user.eoc27/02/2004

<?php

#-----
# Copyright 2004 Earth Observation Centre.
# Commonwealth Scientific and Industrial Research Organisation
#
# Developed as part of the Web-CATS project.
#-----

#-----
#
# Title: check_user
#
# Path: sessions/check_user
#
# Purpose: Reports outcome of authentication to user. If successful,
# provides opportunity to specify a new job name or to select
# from a list of previous jobs.
#
# Inputs: username and password
#
# Outputs: loadjob (new/old) & either j_name (for new) or job (for
# existing)
#
# Predecessors: sessions/login
#
# Successors: sessions/check_job
#
# Environment: N/A
#
# Side-effects: None
#-----

session_start();

include "../parameters.inc";
include "../web-cats.$script";
include "login.class";

print_page_header('Login Results', 'Web-CATS User Login');
print_standard_links();

# Get username and password from login page
extract($_REQUEST);

$c = new login;

# Database name and login details come from parameters.inc
$c->connect($webcats_database, $webcats_host,
           $webcats_username, $webcats_passwd);

$q = "SELECT * FROM user_details WHERE name='$username' AND password='$password'";
$result = $c->db_query($q);

# WARNING: Explicit MYSQL dependency here
$sql_array = mysql_fetch_array($result, MYSQL_ASSOC);

```

check_user.eoc

27/02/2004

```

if ($debug && 0)
{
    print "<pre>";
    print_r($sql_array);
    print "</pre>\n";
}

if ($sql_array[user_status]==0)
{
    # Fail
    echo "<br>Invalid credentials, hit your browsers 'Back' button and try again ";
    echo "or contact the Web-CATS team to request authorisation.<br><br><br>";
}
else
{
    # Success - record username in session variable
    echo "<h2>Welcome $sql_array[name]</h2>";
    $_SESSION['username'] = $sql_array[name];

    print <<< END
    You can either:<br>
    <ol>
    <li>Create a new processing job from scratch;</li>
    <li>Copy the spatial-search attributes of a previous job.</li>
    </ol>

    <Form method=post action=check_job.$script>
    <Table border=1>
    <tr>
    <td><input NAME='loadjob' type='Radio' VALUE='new' checked>New</td>
    <td><input NAME='loadjob' type='Radio' VALUE='old' >Existing</td>
    </tr>
    <tr>
    <td>
    New job name:<br><input name='j_name' type='text'></td><td>
    END;

    # Get list of jobs for this user (sorted alphabetically)
    $q = "SELECT jobname FROM job_details WHERE name='$username' ORDER BY jobname;";
    $result2 = $c->db_query($q);
    $sql_array2 = $c->sql_to_array($result2);

    #print "<pre>";
    #print_r($sql_array2);
    #print "</pre>";

    echo "<center>";
    echo "<SELECT name='job' size=5>\n";

    $job = "";
    for ($i=0;$i<count($sql_array2);$i++)
    {
        if ($sql_array2[$i][jobname] != $job)
        {
            $job = $sql_array2[$i][jobname];
            echo "<OPTION VALUE='$job'>$job</OPTION>\n";
        }
    }
}

```

check_user.eoc

27/02/2004

```
    } }
    print <<< END
</select>
</center>
<br>
</td></tr></table>
<br><br><input type='submit' name='Submit' value='Submit'>
</form>
END;
}
$c->close;
print_page_trailer();
?>
```

Figure 8.2: PHP script check_user.eoc.

8.3 C - Database tables

Version: 20040107140143

Key: **Primary Key**; *Foreign Key*; Indexed*

Table:		avhrr_hrpt_scene
scene_id*	int	Unique identifier for this file
crejld	real	JD this record was created
modjld	real	JD this record last modified
version	int	Version no. of this record
sat_id*	int	Pointer to satellite table
stn_id*	int	Pointer to station table
start_jd*	real	JD at start of pass
end_jd*	real	JD at end of pass
start_orbit*	int	Orbit no. at start of pass
end_orbit	int	Orbit no. at end of pass
start_dir	varchar(1)	Direction (A/D) at start of pass
centre_dir	varchar(1)	Direction (A/D) at middle of pass
end_dir	varchar(1)	Direction (A/D) at end of pass
bands	varchar(20)	List of bands present
num_dropped	int	No. of dropped lines
daynight	varchar(1)	Night or day (N/D) flag
sunzen	real	Solar zenith angle at image centre (degrees)
nwlat	real	Lat of NW point (deg)
nwlon	real	Lon of NW point (deg)
nnlat	real	Lat of N nadir point (deg)
nnlon	real	Lon of N nadir point (deg)
nelat	real	Lat of NE point (deg)
nelon	real	Lon of NE point (deg)
cwlat	real	Lat of centre W point (deg)
cwlon	real	Lon of centre W point (deg)
cnlat	real	Lat of centre nadir point (deg)
cnlon	real	Lon of centre nadir point (deg)
celat	real	Lat of centre E point (deg)
celon	real	Lon of centre E point (deg)
swlat	real	Lat of SW point (deg)
swlon	real	Lon of SW point (deg)
snlat	real	Lat of S nadir point (deg)
snlon	real	Lon of S nadir point (deg)
selat	real	Lat of SE point (deg)
selon	real	Lon of SE point (deg)
eqcrs	real	Equatorial crossing (deg)
satvw	int	Satellite view flag (1=sat view, -1=non-sat)
clkcor	int	Satellite Clock correction
dtime	real	Dtime
dalt	real	Dalt
ephem	varchar(80)	ADAPS Ephemeris Information
roll1	real	First component of Roll
roll2	real	Second component of Roll
roll3	real	Third component of Roll
roll4	real	Fourth component of Roll
roll5	real	Fifth component of Roll
pitch1	real	First component of Pitch
pitch2	real	Second component of Pitch
pitch3	real	Third component of Pitch
pitch4	real	Fourth component of Pitch
pitch5	real	Fifth component of Pitch
yaw1	real	First component of Yaw
yaw2	real	Second component of Yaw
yaw3	real	Third component of Yaw
yaw4	real	Fourth component of Yaw
yaw5	real	Fifth component of Yaw
flags	varchar(10)	Scene Status Flag

8 Appendix

Table: browse		
<i>input_file_id*</i>	int	Unique id. of input file
<i>output_file_id*</i>	int	Unique id. of output file

Table: file		
file_id*	int	Unique id. of file
name*	varchar(80)	Name for file
<i>media_id*</i>	int	Media Id.
location*	varchar(80)	File location on media (path/file seq no.)
<i>file_fmt_id</i>	int	Id. of file format in file_fmt table
processing_type	varchar(40)	Processing Type (eg [time—spatial]_mosaic, stitched, fullscene, subset, *_br)
nominal_res	real	Nominal resolution (m)
no_samples	int	Number of samples per line
no_lines	int	Number of lines
no_bands	int	Number of bands
bitpix	int	No. of bits per pixel
size	real	Size (Mb)
sensor_table	varchar(40)	Table holding sensor specific scene description
<i>scene_id*</i>	int	Scene identifier within sensor-specific table
log_exists	int	File log flag (0/1)
crejd	real	JD this record was created
modjd	real	JD this record last modified
version	int	Version no. of this record

Table: file_alias		
<i>file_id*</i>	int	Unique id. of file
pathname*	varchar(80)	Full path name of file on disk

Table: file_fmt		
file_fmt_id*	int	Unique identifier for this format
type	varchar(40)	Description of file type in the form <software>.<organisation>
supplementary_name	varchar(40)	Name of supplementary table (if any)
supplementary_id	int	Id. of format in supplementary table

Table: file_log		
<i>file_id*</i>	int	Unique identifier for this file
log_url	varchar(80)	URL of log file
description	varchar(80)	Description of operation in log
crejd	real	JD this record was created

Table: file_transform_trace		
<i>input_file_id*</i>	int	Unique id. of input file
<i>output_file_id*</i>	int	Unique id. of output file
transform	varchar(80)	Relationship of output to input, eg - program used or operation applied

Table: loaded_stn_media		
<i>media_id*</i>	int	Media Id. of loaded tape

8.3 C - Database tables

Table: media_copies		
<i>input_media_id*</i>	int	Unique id. of input media
<i>output_media_id*</i>	int	Unique id. of output media
relationship	varchar(80)	Relationship of output to input, eg - identical, composite, child/parent

Table: media_fmt		
media_fmt_id*	int	Unique identifier for this media format
media_fmt_name*	varchar(20)	Unique name for this format
media_fmt_string	varchar(80)	Description of media format, eg TAPE=SS(HD)+ or DISK

Table: ops_log		
log_url	varchar(80)	URL of log file
description	varchar(80)	Description of operation in log
crejd*	real	JD this record was created

Table: scene_transform_trace		
<i>input_scene_id*</i>	int	Unique id. of input scene
<i>output_scene_id*</i>	int	Unique id. of output scene
transform	varchar(80)	Relationship of output to input, eg - program used or operation applied

Table: spacecraft		
sat_name*	varchar(40)	Spacecraft name
sat_alias*	varchar(40)	Alternate spacecraft name
sat_id*	int	Local EOC spacecraft identifier
alt_id*	varchar(40)	Id ascribed by operating agency
launch_date	real	Launch date
first_data	real	Date of first data
last_data	real	Date of last data
note	varchar(80)	Additional note

Table: station		
stn_id*	int	Unique identifier for this station
ceos*	varchar(4)	CEOS ¹ station code
lat	real	Station latitude
lon	real	Station longitude
x	real	WGS72 ² X-position (m)
y	real	WGS72 Y-position (m)
z	real	WGS72 Z-position (m)
crejd	real	JD this record was created
modjd	real	JD this record last modified
version	int	Version no. of this record
name	varchar(40)	Station Name/Location

¹ CEOS = Committee on Earth Observation Satellites

² WGS=World Geodetic System

Table: stn_file_fmt		
fmt_id*	int	Unique identifier for this format
fmt_type*	varchar(20)	Name of this format
description	varchar(80)	Brief description or URL of description
program	varchar(40)	Name of a program to read the format
start_date	real	Date of first data
end_date	real	Date of last data
<i>stn_id*</i>	int	Reception station identifier

8 Appendix

Table:		storage
media_id*	int	Unique id. of media
name*	varchar(80)	Label
media_type	varchar(40)	Media type - 8mm, 4mm, DLT, CD, 3.5floppy, 5.25floppy, 9track
location	varchar(80)	Media physical storage location
content_type	varchar(80)	Type of files on media (stitched, stn. raw..)
earliest_file*	real	Date and time of earliest file on media
latest_file*	real	Date and time of latest file on media
no_of_files	int	Number of files stored on tape
mb_on_storage	real	Mb. of data on media
capacity	real	Capacity of media (Mb)
write_style	varchar(40)	Write format of data on tape (dd/tar)
note	varchar(80)	Operator note
blocking	int	File blocking factor on tape
hw_compression	int	Hardware compression flag (0/1)
sw_compression	int	Software compression flag (0/1)
media_fmt_id	int	Media format identifier
stn_id	int	Receiving or processing station Id.
log_exists	int	Media log flag (0/1)
crejd	real	JD this record was created
modjd	real	JD this record last modified
version	int	Version no. of this record

Table:		storage_log
media_id*	int	Unique identifier for this media
log_url	varchar(80)	URL of log file
description	varchar(80)	Description of operation in log
crejd	real	JD this record was created

Table:		subset
input_file_id*	int	Unique id. of input file
output_file_id*	int	Unique id. of output file