High Energy Physics Libraries Webzine

Issue 5 / November 2001

http://library.cern.ch/HEPLW/5/papers/1

The NASA Astrophysics Data System: Free Access to the Astronomical Literature On-line and through Email

Guenther Eichhorn, Alberto Accomazzi, Carolyn S. Grant, Michael J. Kurtz and Stephen S. Murray (*)

28/09/2001

Abstract:

The Astrophysics Data System (ADS) provides access to the astronomical literature through the World Wide Web. It is a NASA funded project and access to all the ADS services is free to everybody world-wide.

The ADS Abstract Service allows the searching of four databases with abstracts in Astronomy, Instrumentation, Physics/Geophysics, and the LANL Preprints with a total of over 2.2 million references. The system also provides access to reference and citation information, links to on-line data, electronic journal articles, and other on-line information. The ADS Article Service contains the full articles for most of the astronomical literature back to volume 1. It contains the scanned pages of all the major journals (Astrophysical Journal, Astronomical Journal, Astronomy & Astrophysics, Monthly Notices of the Royal Astronomical Society, and Solar Physics), as well as most smaller journals back to volume 1. The ADS can be accessed through any web browser without signup or login. Alternatively an email interface is available that allows our users to execute queries via email and to retrieve scanned articles via email. This might be interesting for users on slow or unreliable links, since the email system will retry sending information automatically until the transfer is complete.

There are now 9 mirror sites of the ADS available in different parts of the world to improve access. The ADS is available at:

http://ads.harvard.edu/

1 Introduction

The NASA Astrophysics Data System Abstract Service is by now a central facility of bibliographic research in astronomy. In a typical month (March 2001) it is used by more than 50,000 individuals, who make ~ 800,000 queries, retrieve ~ 28,000,000 bibliographic entries, read ~ 600,000 abstracts and ~ 130,000 articles, consisting of ~ 1,100,000 pages. The ADS is a key element in the emerging digital information resource for astronomy, which has been dubbed Urania ([Boyce 1996]). The ADS is tightly interconnected with the major journals of astronomy, and the major data centers. A detailed description of the ADS has been published in a special issue of Astronomy & Astrophysics Supplements in April, 2000 (Overview: [Kurtz et al. 2000], Search Engine and User Interface: [Eichhorn, et al. 1999], System Architecture: [Accomazzi et al. 2000], Data: [Grant et al. 2000]).

The first major part of the ADS is the Abstract Service. It was started in 1993 with a custom-built networking software system to provide access to distributed data ([Murray et al. 1992]). By summer 1993 a connection had been made between the ADS and SIMBAD (Set of Identifications, Measurements and Bibliographies for Astronomical Data, [Egret et al. 1991]) at the Centre des Données de Strasbourg (CDS), permitting users to combine natural language subject matter queries with astronomical object name queries ([Grant, Kurtz, & Eichhorn 1994]). The user interface for this first version of the ADS was built with the custom-built software system that the ADS used at that time. The search engine of this first implementation used a commercial database system. A description of the system at that time is in [Eichhorn 1994a].

By early 1994 The World Wide Web (WWW, [http://doc.cern.ch/heplw/5/papers/1/geichhorn.html#www]) had matured and was widely accessible through the NCSA Mosaic Web Browser ([Schatz & Hardin 1994]). It now was possible to make the ADS Abstract Service available via a web forms interface; this was released in February 1994. Within five weeks of the initial WWW release use of the Abstract Service quadrupled (from 400 to 1600 users per month), and it has continued to rise ever since ([Eichhorn 1997a]). The WWW interface to the ADS is described by [Eichhorn et al. 1995b] and [Eichhorn et al. 1995a].

The second major part of the ADS is the Article Service. It contains scanned full journal articles for most of the astronomical journal literature going back to volume 1 for most journals. The first full article bitmaps, which were of Astrophysical Journal Letters articles, were put on-line in December 1994 ([Eichhorn et al. 1994b]). By the summer of 1995 the bitmaps were current and complete going back ten years. At that time the Electronic ApJ Letters (EApJL) ([Boyce 1995]) went on-line. From the start the ADS indexed the EApJL, and pointed to the electronic version. Also from the beginning the reference section of the EApJL pointed (via WWW hyperlinks) to the ADS abstracts for papers referenced in the on-line articles.

With time, other interfaces to the abstracts and scanned articles were developed to provide other information providers the means to integrate ADS data into their system ([Eichhorn et al. 1996b]).

With the adoption of the WWW user interface and the development of the custom-built search engine, the current version of the ADS Abstract Service was basically in place. Currently the ADS system consists of four semi-autonomous (to the user) abstract services covering Astronomy/Planetary Sciences, Instrumentation, Physics, and Astronomy Preprints. Combined there are over 2.3 million abstracts and bibliographic references in the system. The Astronomy

Service is by far the most advanced, and accounts for ~ 85% of all ADS use ([Kurtz et al. 2000], [Eichhorn, et al. 1999]).

The following sections will describe the data (section 2), the user interface (section 3), the user customization capabilities (section 4), the search engine (section 5), the mirroring system (section 6), some query examples (section 7), and finally some access statistics and access patterns (section 8).

2 Data

This section describes the data holdings in the abstract and article service of the ADS as well as the links database.

2.1 Abstracts

The abstracts in the ADS come from many different sources (see [Grant et al. 2000]). The original set came from the NASA STI database. We now receive basic bibliographic information (title, author, page number) from essentially every journal of astronomy. Most publishers also send us abstracts, while some who cannot send abstracts, allow us to scan their journals. For these journals we build abstracts through optical character recognition (OCR). Finally we receive abstracts from the editors of conference proceedings, and from individual authors.

As of Sept, 2001 there are ~ 672,000 astronomy references indexed in the ADS, the database is nearly complete for the major journals articles beginning in 1975. In the Physics database there are ~1.1 million references, and in the Instrumentation database there are ~ 605,000 references. Approximately half of all references have abstracts, the other half only have titles, authors, and journal information.

2.2 Bitmaps

The ADS has obtained permission to scan, and make freely available on-line, page images of the back issues of all the major journals and most smaller journals in astronomy. In most cases the bitmaps of current articles are put on-line after an embargo period, to protect the financial integrity of the journal.

We plan to provide for each collaborating journal, in perpetuity, a database of page images (bitmaps) from volume 1 page 1 to the first issue which the journal considers to be fully on-line as published. This will provide (along with the indexing and the more recent archives held by the journals) a complete electronic digital library of the major literature in astronomy.

On a longer term we plan to scan old observatory reports and defunct journals, to finally have a full historical collection on-line. This work is beginning with a collaboration with the Wolbach Library at the Harvard-Smithsonian Center for Astrophysics and the Harvard Preservation Project ([Eichhorn et al. 1997b]; [Corbin & Coletti 1995]).

The first journal to be scanned was the *ApJ Letters* in January, 1995. By now there are ~ 1.2 million scanned pages on-line in the ADS in ~ 130,000 articles. The bitmaps in the ADS have been scanned at 600 dpi using a high speed scanner and generating a 1 bit/pixel monochrome image for each page (see [Grant et al. 2000]). The files created are then automatically processed in order to de-skew and center the text in each page, resize images to a standard U.S. Letter size (8.5 x

11 inches), and add a copyright notice at the bottom of each page. Adding the copyright notice on each page is important, since the ADS makes it very easy to reprint individual pages. Such individual pages would lose the information on where they came from and who owns the copyright for them. For each original scanned page, two separate image files of different resolutions are generated and stored on disk. The availability of different resolutions allows users the flexibility of downloading either high or medium quality documents, depending on the speed of their Internet connection. The image formats and compression used were chosen based on the available compression algorithms and browser capabilities. The high resolution files currently used are 600 dpi, 1 bit/pixel TIFF (Tagged Image File Format) files, compressed using the CCITT Group 4 facsimile encoding algorithm. The medium resolution files are 200 dpi, 1 bit/pixel TIFF files, also with CCITT Group 4 facsimile compression.

2.3 Links

The ADS responds to a query with a list of references and a set of hyperlinks showing what data are available for each reference (see [Eichhorn, et al. 1999]). There are ~ 2 million hyperlinks in the ADS, of which ~ 40% are to sources external to the ADS project.

The largest number of external links are to SIMBAD, the NASA Extragalactic Database (NED), the Space Telescope Science Institute (STScI), and the electronic journals. A rapidly growing number, although still small in comparison to the others, are to data tables created by the journals and maintained by the CDS and the Astronomical Data Center (ADC) at Goddard. These links are an extremely important aspect of the ADS.

Table 1 shows the links that we currently provide when available.

Table 1: Link types in the ADS database.

A	Abstract	Full abstract of the article. These abstracts come from different sources.
C	Citations	A list of articles that cite the current article. This list is not necessarily complete (see `R' References).
D	On-line Data	Links to on-line data at other data centers.
E	Electronic Article	Links to the on-line version of the article. These on-line versions are in HTML format for viewing on-screen, not for printing.[a]
F	Printable Article	Links to on-line articles in PDF or Postscript format for printing.[a]
G	Gif Images	Links to the images of scanned articles in the ADS Article Service.
I	Author Comments	Links to author supplied additional information (e.g. corrections, additional references, links to data),
L	Library Entries	Links to entries in various library on-line systems.
M	Mail Order	Links to on-line document delivery systems at the publisher/owner of the article.
N	NED Objects	Access to lists of objects for the current article in the NED database.
O	Associated Articles	A list of articles that are associated with the current article. These can be errata or other articles in a series.
P	Planetary	Links to datasets at the Planetary Data System.

	Data System	
R	References	A list of articles referred to in the current article. For older articles these lists are not necessarily complete, they contain only references to articles that are in the ADS database. For some articles that are on-line in electronic form, the `R' link points to the on-line reference list, and therefore the complete list of references in that article.[a]
S	SIMBAD Objects	Access to lists of objects for the current article in the SIMBAD database.
T	Table of Contents	Links to the list of articles in a books or proceedings volume.
U	Also-Read Articles	Links to the list of articles that were read by the same people that read the current article.

[a]There is generally access control at the site that serves these on-line articles

A more detailed description of resources in the ADS that these links point to is provided in [Grant et al. 2000].

Some of these links (for instance the `D' links) can point to more than one external information provider. In such cases the link points to a page that lists the available choices of data sources. The user can then select the more convenient site for that resource, depending on the connectivity between the user site and the data site.

2.4 Citations and References

The use of citation histories is a well known and effective tool for academic research ([Garfield 1979]). In 1996 the American Astronomical Society purchased a subset of the Science Citation Index from the Institute for Scientific Information, to be used in the ADS; this was updated in 1998. This subset only contains references which were already in the ADS, thus it is seriously incomplete in referring to articles in the non-astronomical literature. This citation information from ISI spans January 1982-September 1998.

The electronic journals all have machine readable, web accessible, reference pages. The ADS points to these with a hyperlink where possible. Several publishers allow us to use these to maintain citation histories; we do this using our reference resolver software. The same software is also used by some publishers to check the validity of their references, pre-publication.

Additionally we use optical character recognition to create reference and citation lists for the historical literature, after it is scanned ([Demleitner, et al. 1999]). This process has handled over 10 million references and added over 6 million parsed references to the ADS citation database.

2.5 Collaboration with CDS/SIMBAD

The CDS has long maintained several of the most important data services for astronomy (e.g. [Jung 1971]; [Jung, Bischoff, & Ochsenbein 1973]; [Genova et al. 1998]); access to parts of the CDS data via ADS is a key feature of the ADS.

ADS users are able to make joint queries of the ADS bibliographic database and the CDS/SIMBAD bibliographic data base. When SIMBAD contains information on an object which is referred to in a

paper whose reference is returned by ADS then ADS also returns a pointer to the SIMBAD data. When a paper has a data table which is kept on-line at the CDS the ADS returns a pointer to it. The CDS-ADS collaboration is at the heart of Urania, a world-wide collaboration of astronomical data providers. More recently the ADS has entered into a collaboration with the NASA Extragalactic Database (NED; [Helou & Madore 1988], [Madore et al. 1992]) which is similar to the SIMBAD portion of the CDS-ADS collaboration.

3 User Interface

The ADS services can be accessed through various interfaces (see [Eichhorn, et al. 1999]). Some of these interfaces use WWW based forms, others allow direct access to the database and search system through Application Program Interfaces (APIs). This section describes the various interfaces and their use, as well as the returned results.

3.1 Forms Based Interfaces

3.1.1 Search Forms

The main query forms (figures 1, 2, 3) provide access to the different abstract databases. These forms are generated on demand by the ADS software. This allows the software to check the user identification through the HTTP (HyperText Transfer Protocol) cookie mechanism (see section 4), so that the software can return a customized query form if one has been defined by the user. It also adapts parts of the form according to the capabilities of the user's web browser.

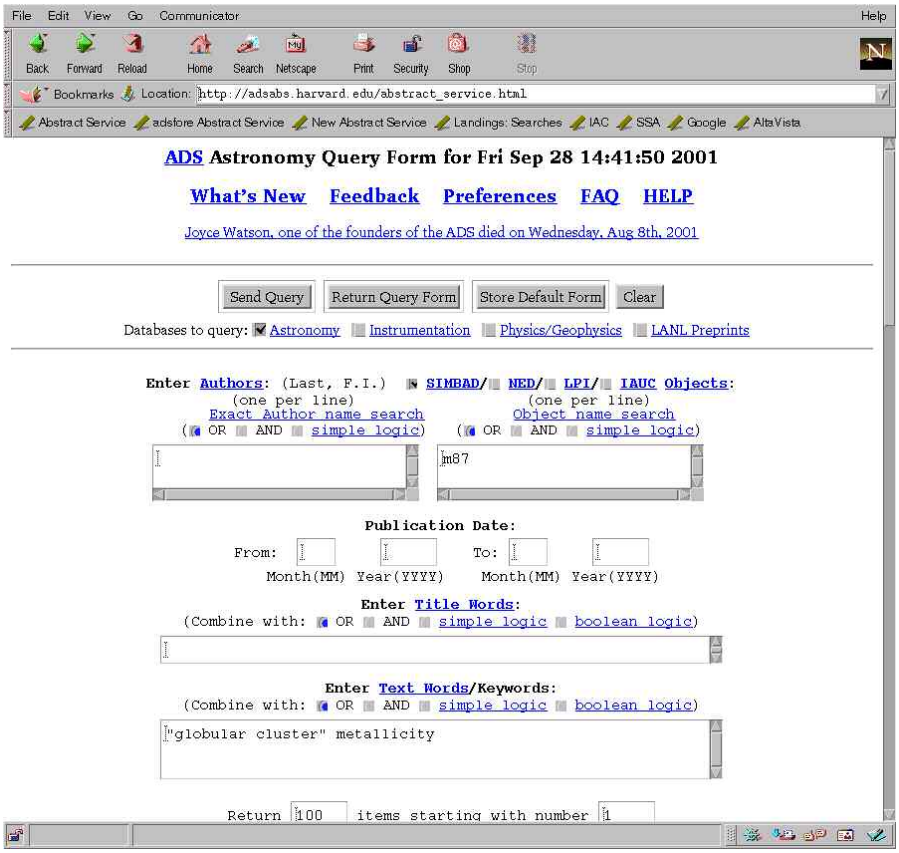

Figure 1: A query to the ADS Abstract Service requesting a listing of papers on the metallicity of M87 globular clusters. SIMBAD, NED, the ADS phrase index, the ADS word index and the ADS

synonym list are all queried, the results are combined and the list shown in figure 4 is returned.

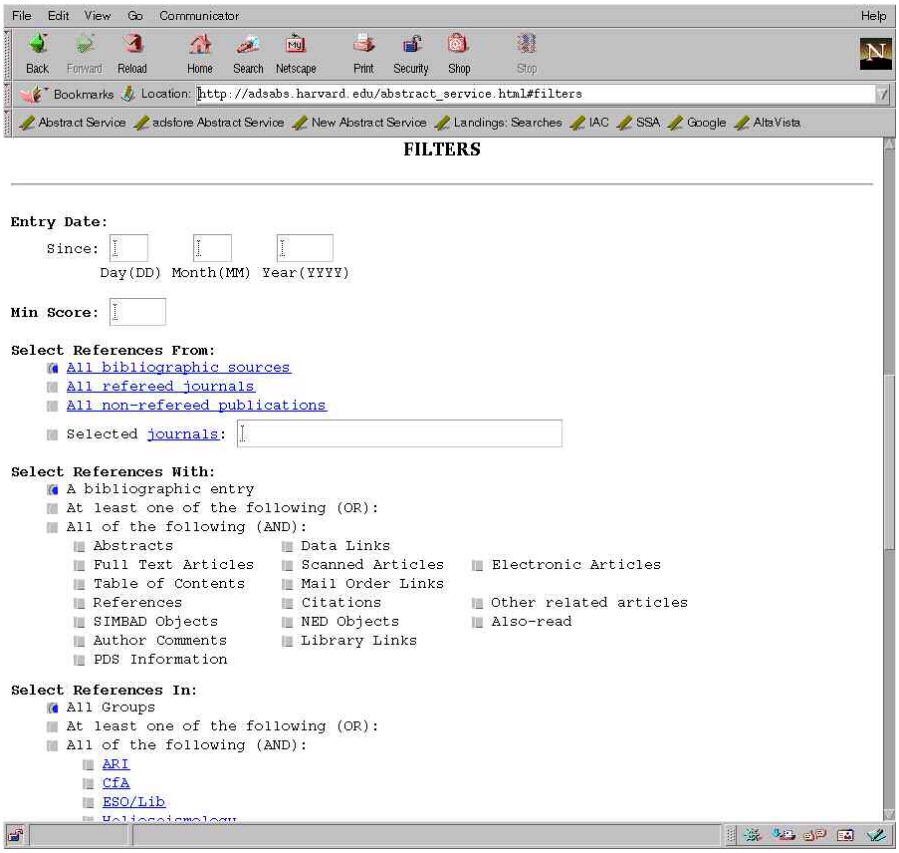

Figure 2: The Filter section of the query form allows selection of references that have specific properties.

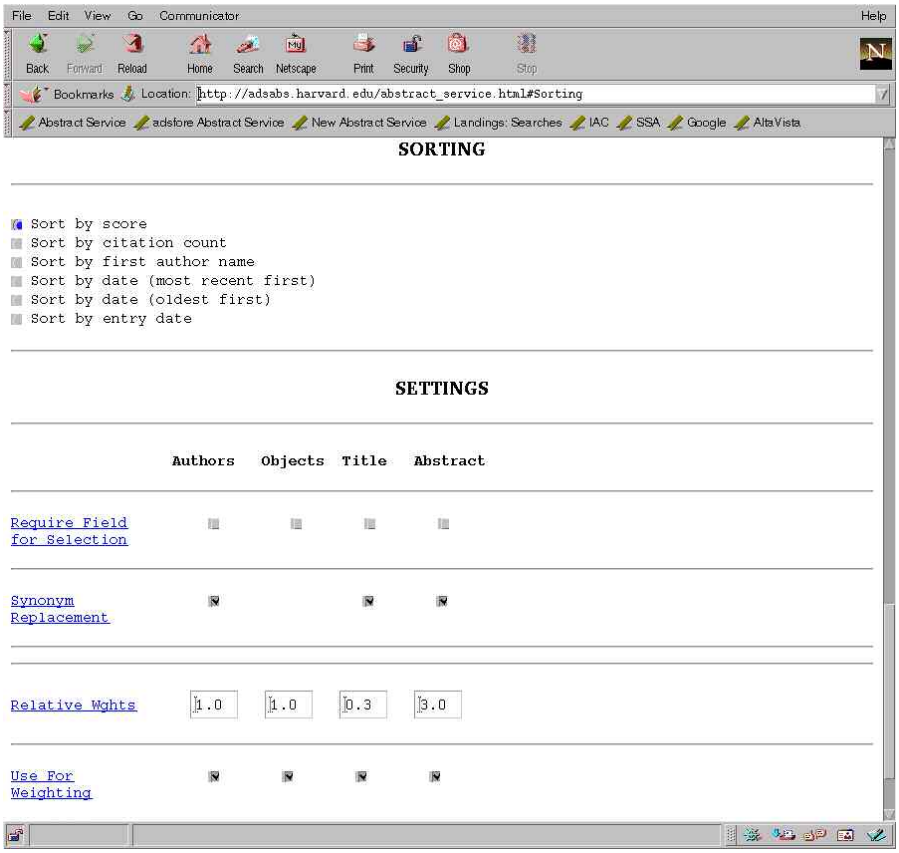

Figure 3: The Settings section of the query form allows the user to customize the search.

The main query form allows the user to specify search terms in different fields. The input parameters in each query field can be combined in different ways, as can the results obtained from the different fields (figure 1). The user can specify how the results are combined through settings on the query form (figure 3). The combined results can then be filtered according to various criteria (figure 2).

The database can be queried for author names, astronomical objects names, title words, and words in the abstract text. References can be selected according to the publication date. The author name, title, and text fields are case insensitive. The object field is case sensitive when the IAU (International Astronomical Union) Circulars object name database is searched, since the IAU object names are case sensitive. In the author and object name fields, the form expects one search term per line since the terms can contain blanks. In the title and text fields line breaks are not significant.

3.1.1.1 Author Name Field

The author names are indexed by last name and by a combination of last name and first initial, separated by a comma. To account for differences in the spelling of the same author name, the search system contains a list of author names that are spelled differently but are in fact names of the same author. This allows for automatically retrieving all versions of common spelling differences. This is useful for instance for German umlaut spelled as Muller and Mueller, or variations in the transliteration of names from non-English alphabets like Cyrillic. An example of such an entry in the author synonym list is:

```
AFANASJEV, V
AFANAS'EV, V
AFANAS'IEV, V
AFANASEV, V
AFANASYEV, V
AFANS'IEV, V
AFANSEV, V
```

Without this synonym replacement capability, author searches would obviously be much less effective. On user request we also include name changes (e.g. due to marriage) in the author synonym list.

Author names are quite often spelled differently in different publications. First names are sometimes spelled out, sometimes only first initials are given, and sometimes middle initials are left out. This makes it impossible to index all different spellings of a name together automatically.

To handle these different requirements, author names are indexed three times, once with the last name only, once with the last name and first initial, and once with the complete name as it is specified in the article.

To access these different indexes, we provide two user interfaces for author queries. The regular user interface allows the user to search for either a last name or a last name combined with the first initial. This allows for fairly discriminating author searches. It is a compromise between the need to discriminate between different authors, and the need to find all instances of a given author. It identifies all different versions of a given author quite reliably, but it indexes together different authors with the same first initial. For cases where this search method is not discriminating enough, we provide a second user interface to the index of the full names, which does not attempt to index different spellings of the same author together. When the user selects ``Exact Author Search'' and specifies an author's last name or last name and first initial, a form is returned with all distinct full

author names that match the specified name. The user then selects all the different spellings of the desired name and queries the database for articles that contain any one of these different versions of an author's name. For instance specifying:

```
Eichhorn, G
```

in the exact author name form returns the list:

```
EICHHORN, G.
EICHHORN, GERHARD
EICHHORN, GUENTHER
EICHHORN, GUNTHER
```

Selecting the first, third, and fourth author name from that list will return all articles by the first author of this article. Any articles by the second author containing only the first initial will also be returned, but this is unavoidable.

3.1.1.2 Object Name Field

This field allows the user to query different databases for references with different astronomical objects. The databases that provide object information are: CDS/SIMBAD, France; the NASA Extragalactic Database (NED) at the Infrared Processing and Analysis Center (IPAC), Jet Propulsion Laboratory (JPL), Pasadena, CA ([Madore et al. 1992]); the IAU Circulars (IAUC) and the Minor Planet Electronic Circulars (MPEC), both provided by the Central Bureau for Astronomical Telegrams (CBAT) at the Harvard-Smithsonian Center for Astrophysics in Cambridge, MA ([Marsden 1980]); and a database with objects from publications from the Lunar and Planetary Institute (LPI) in Houston (mainly Lunar sample numbers and meteorite names). The user can select which of these databases should be queried. If more than one database is searched, the results of these queries are merged. The LPI database does not have any entries in common with the other databases. The SIMBAD, NED, and IAUC databases sometimes have information about the same objects.

3.1.1.3 Title and Abstract Text Fields

These fields query for words in the titles of articles or books, and in the abstracts of articles or descriptions of books respectively. The words from the title of each reference are also indexed in the text field so they will be found through either a title or a text search. Before querying the database the input in these fields is processed as follows:

1. Apply translation rules. This step merges common expressions into a single word so that they are searched as one expression. Regular expression matching is used to convert the input into a standard format that is used to search the database. For instance *M 31* (with a space) is translated to *M31* (without a space) for searching as one search term. In order to make this general translation, a regular expression matching and substitution is performed that translates all instances of an `M' followed by one or more spaces or a hyphen followed by a number into `M' directly followed by the number. Other translation rules include the conversion of *NGC 1234* to *NGC1234*, contractions of *T Tauri*, *Be Star*, *Shoemaker Levy*, and several others (see [Accomazzi et al. 2000]).

2. Remove punctuation. In this step all non-alphanumeric characters are removed, unless they are significant (for instance symbols used in the simple logic (see below), `+' and `-' before numbers, or `.' within numbers).

3. Translate to uppercase. All information in the index files is in uppercase, except for object names from the IAU Circulars.

4. Remove stop words. This step removes all non-significant words. This includes words like `and', `although', `available', etc (for more details see [Accomazzi et al. 2000]).

In the title and text fields, searching for phrases can be specified by enclosing several words in either single or double quotes, or concatenating them with periods (`.') or hyphens (`-'). All these accomplish the same goal of searching the database for references that contain specified sequences of words. The database is indexed for two-word phrases in addition to single words. Phrases with more than two words are treated as a search for sets of two-word phrases containing the first and second word in the first phrase, the second and third word in the second phrase, etc.

3.1.2 Searching

After the search terms are pre-processed, the databases of the different fields are searched for the resulting list of words, the results are combined according to the selected combination rules, and the resulting score is calculated according to the selected scoring criteria. These combination rules provide the means for improving the selectivity of a query.

3.1.2.1 Search Word Selection

The database is searched for the specified words as well as for words that are synonymous with the specified term. One crucial part to successful searches in a free text search system is the ability to not only find words exactly as specified, but also similar words. This starts with simply finding singular and plural forms of a word, but then needs to be extended to different words with the same meaning in the normal usage of words in a particular field of science. In Astronomy for instance ``spectrograph" and ``spectroscope" have basically the same meaning and both need to be found when one of these words is specified in the query. Even further reaching, more discipline-specific synonyms are necessary for efficient searches such as ``metallicity" and ``abundance" which have the same meaning in astronomical word usage. In order to exhaustively search the database for a given term, it is important to search for all synonyms of a given word. The list of synonyms was developed manually by going through the list of words in the database and grouping them according to similar meanings. This synonym list is a very important part of the ADS search system and is constantly being improved (see [Accomazzi et al. 2000]).

The list of synonyms also contains non-English words associated with their English translations. These words came from non-English reference titles that we included in the database. This allows searches with either the English or non-English words to find references with either the English word or the non-English translation. We are in the process of extending this capability by including translations of most of the words in our database into several languages (German, French, Italian, Spanish). This will allow our users to phrase queries in any of these languages.

By default a search will return references that contain the search word or any of its synonyms. The user can choose to disable this feature if for some reason a specific word needs to be found. The synonym replacement can be turned off completely for a field in the ``Settings" section of the query form. This can be used to find a rare word that is a synonym of a much more frequent word, for instance if you want to look for references to ``dateline", which is a synonym to ``date". Synonym replacement can also be enabled or disabled for individual words by prefixing a word with `=' to force an exact match without synonym replacement. When synonym replacement is disabled for a field, it can be turned on for a particular word by prefixing it with `#'.

3.1.2.2 Selection Logic Within a Field

There are four different types of combinations of results for searches within a field possible.

1. OR
2. AND
3. Simple logic
4. Full boolean logic

1. Combination by `OR': The resulting list contains all references that contain at least one of the search terms.

2. Combination by `AND': The resulting list has only references that contain every one of the search terms.

3. Combination by simple logic: The default combination in this logic is by `OR'. Individual terms can be either required for selection by prefixing them with a `+', or can be selected against by prefixing them with a `-'. In the latter case only references that do not contain the search term are returned. If any of the terms in the search is prefixed by a `+', any other word without a prefix does not influence the resulting list of references. However, the final score (see below) for each reference will depend on whether the other search terms are present.

4. Combination by full boolean logic: In this setting, the user specifies a boolean expression containing the search terms and the boolean operators `and', `or', and `not', as well as parentheses for grouping. A boolean expression could for instance look like:

(pulsar or ``neutron star'') and (``red shift'' distance) and not 1987A

This expression searches for references that contain either the word pulsar or the phrase ``neutron star'' and either the phrase ``red shift'' or the word distance (``or'' being the default), but not the word 1987A.

3.1.2.3 Selection Logic Between Fields

In the settings part of the query form, the user can specify fields that will be required for selection. If a field is selected as ``Required for Selection'' only references that were selected in the search specified in that field will be returned. If one field is selected as ``Required for Selection'', the searches in fields that are not set as ``Required for Selection'' do not influence the resulting list, but they influence the final score.

3.1.3 Scoring

The list of references resulting from a query is sorted according to a ``score'' for each reference. This score is calculated according to how many of the search items were matched. The user has the choice between two scoring algorithms:

1. proportional scoring
2. weighted scoring

These scoring algorithms have been analyzed by [Salton & McGill 1983].

In proportional scoring, the score is directly proportional to the number of terms found in the reference. In weighted scoring, the score is proportional to the inverse logarithm of the frequency of the matched word. This weighting gives higher scores for words that are less frequent in the database and therefore presumably more important indicators of the relevance of a match. In the settings section of the query form the user can select which type of scoring should be used for each query field separately. The default setting for title and text searches is the weighted scoring. For author searches proportional scoring is the default. Once the score for each query field is calculated, the scores are normalized so that a reference that matches all words in a field receives a score of 1.

The normalized scores from the different fields are then combined to calculate a total score. Again the result is normalized so that a reference that matches all words in each query field has a score of 1. The user can influence this combining of scores from the different search fields by assigning weights to the different fields. This allows the user to put more emphasis in the selection process on, for instance, the object field by assigning a higher weight to that field. Another use of the weight field is to select against a field. For instance specifying an object name and an author name and selecting a negative weight to the author field will select articles about that object that were *not* written by the specified author.

The relative weights for the different search fields can be set by the user. The ADS provides default weights as follows:

```
Authors: 1.0
Objects: 1.0
Title:   0.3
Text:    3.0
```

These default weights were determined on theoretical grounds, combined with trial and error experimentation. We used different search inputs from known research fields and different weights and ranked the resulting lists according to how well they represented articles from these research fields. The weights listed above gave the best results.

3.1.4 Filtering of Selected References

The selected references can be filtered according to different criteria (see section 5.5) in order to reduce the number of returned references. The user can select references according to their entry date in the database, a minimum score (see above), the journal they are published in, whether they have pointers to selected external data sources, or whether they belong to one or more of several groups of references. This allows a user for instance to select only references from refereed journals or from one particular journal by specifying its abbreviation. It also allows a user to select only references that have links to external data sets, on-line articles, or that have been scanned and are available through the ADS Article Service.

3.1.5 Display of Search Results

The ADS system returns different amounts of information about a reference, depending on what the user request was. This section describes the different reference formats.

3.1.5.1 Short Reference Display

The list of references returned from a query is displayed in a tabular format. The returned references are sorted by score first. For equal scores, the references are sorted by publication date with the latest publications displayed first.

A typical reference display is shown in figure 4. The fields in such a reference are shown in figure 5. They are as follows:

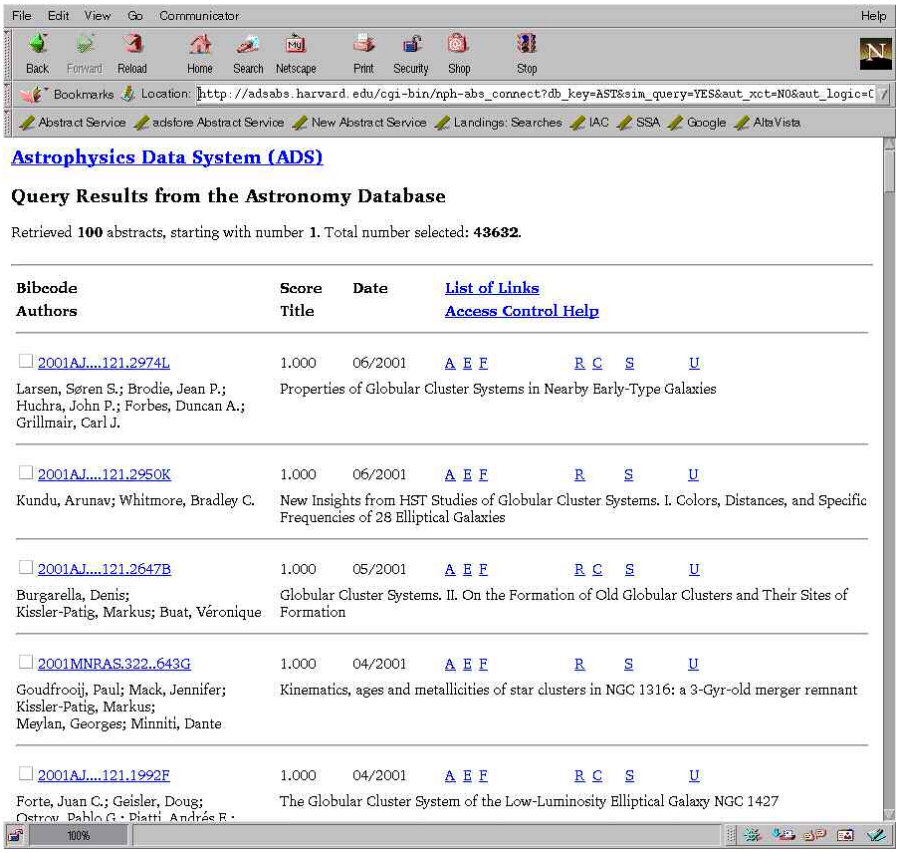

Figure 4: The top of the list ADS returns when the query shown in figure 1 is made.

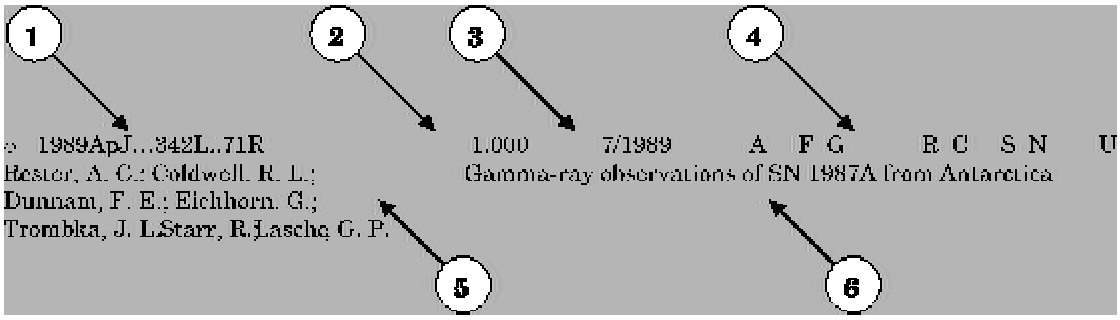

Figure 5: Entries in the list of references returned by an ADS query contain the bibliographic code (1) the matching score (2), the publication date (3), a list of data links (4), the list of authors (5), and the title of the reference (6).

1. Bibliographic Code: This code identifies the reference uniquely (see [Grant et al. 2000] and [Schmitz et al. 1995]). Two important properties of these codes are that they can be generated from a regular journal reference, and that they are human readable and can be understood and interpreted.

2. Score: The score is determined during the search according to how well each reference fits the query.

3. Date: The publication date of the reference is displayed as mm/yyyy.

4. Links: The links are an extremely important aspect of the ADS. They provide access to information correlated with the article (see section 2.3.

5. Authors: This is the list of authors for the reference. Generally these lists are complete. For some of the older abstracts that we received from NASA/STI, the author lists were truncated at 5 or 10 authors, but every effort has been made to correct these abbreviated author lists (see [Grant et al. 2000]).

6. Title: The complete title of the reference.

The reference lists are returned as forms if table display is selected (see section 4). The user can select some or all of the references from that list to be returned in any one of several formats:

i. HTML format: The HTML (HyperText Markup Language) format is for screen viewing of the formatted record.

ii. Portable Format: This is the format that the ADS uses internally and for exchanging references with other data centers. A description of this format is available on-line at:

http://adsabs.harvard.edu/abs_doc/abstract_format.html

iii. BibTeX format: This is a standard format that is used to build reference lists for TeX (a typesetting language especially suited for mathematical formulas) formatted articles.

iv. ASCII format: This is a straight ASCII text version of the abstract. All formatting is done with spaces, not with tabs.

v. User Specified Format: This allows the user to specify in which format to return the reference. The default format for this selection is the bibitem format from the AASTeX macro package. The user can specify an often used format string in the user preferences (see section 4). This format string will then be used as the default in future queries.

The user can select whether to return the selected abstracts to the browser, a printer, a local file for storage, or email it to a specified address.

3.1.5.2 Full Abstract Display

In addition to the information in the short reference list, the full abstract display (see figure 6) includes, where available, the journal information, author affiliations, language, objects, keywords, abstract category, comments, origin of the reference, a copyright notice, and the full abstract. It also includes all the links described above.

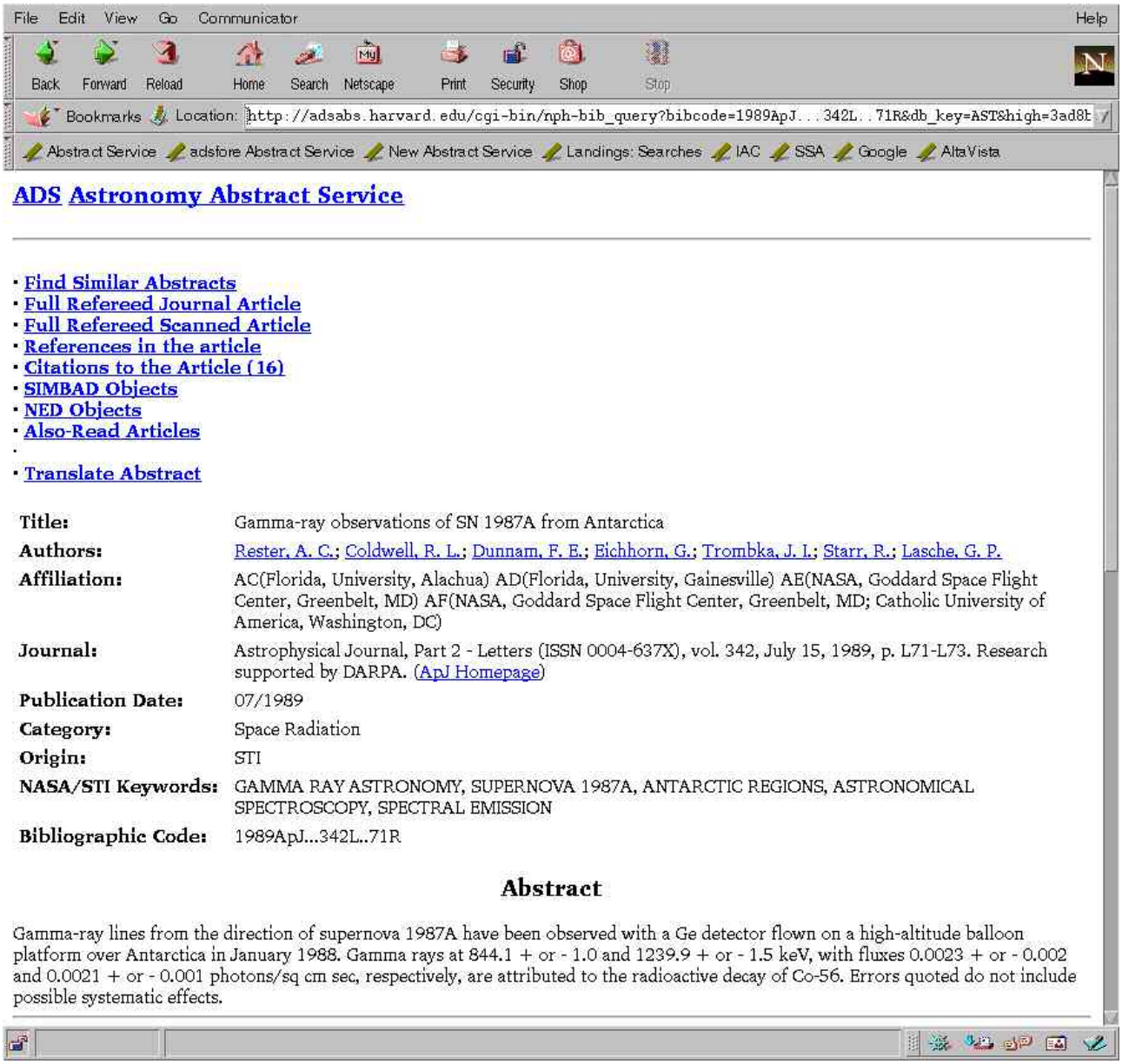

Figure 6: The full abstract display contains (where available) the title, author list, journal information, author affiliations, publication date, keywords, the origin of the reference, the bibliographic code, the abstract, object names, abstract category, and a copyright notice.

For abstracts that are displayed as a result of a search, the system will highlight all search terms that are present in the returned abstract. This makes it easy to locate the relevant parts in an abstract. Since the highlighting is somewhat resource intensive, it can be turned off in the user preference settings (see section 4).

For convenience, the returned abstract contains links that allow the user to directly retrieve the BibTex or the custom formatted version of the abstract.

The full abstract display also includes a form that provides the capability to use selected information from the reference to build a new query to find similar abstracts. The query feedback mechanism makes in-depth literature searches quick and easy. The user can select which parts of the reference to use for the feedback query (e.g. authors, title, keywords, or abstract). The feedback query can either be executed directly, or be returned as a query form for further modification before

executing it, for instance to change the publication date range or limit the search to specific journals. This query feedback mechanism is a very powerful means to do exhaustive literature searches and distinguishes the ADS system from most other search systems. A query feedback ranks the database against the record used for the feedback and sorts it according to how relevant each reference is to the search record. The query feedback can be done across databases. For instance a reference from the Astronomy database can be used as query feedback in the preprint database to see the latest work in the field of this article.

If the article for the current reference has been scanned and is available through the ADS Article Service (see below), printing options are available in the abstract display as well. These printing options allow the printing of the article without having to retrieve the article in the viewer first.

3.1.5.3 Full Article Display

The article display normally shows the first page (figure 7) of an article at the selected resolution and quality (see section 4). The user can select resolutions of 75, 100, or 150 dots per inch (dpi) and image qualities of 1, 2, 3, or 4 bits of greyscale per pixel in the user preferences. These gif images are produced on demand from the stored tiff images (see section 2.2 and [Grant et al. 2000]). The default version of the gif images (100 dpi, 3 bit greyscale) is cached on disk. The cache of these gif images is managed to stay below a maximum size. Any time the size of the cached gif images exceeds the preset cache size, the gif images of pages that have not been accessed recently are deleted.

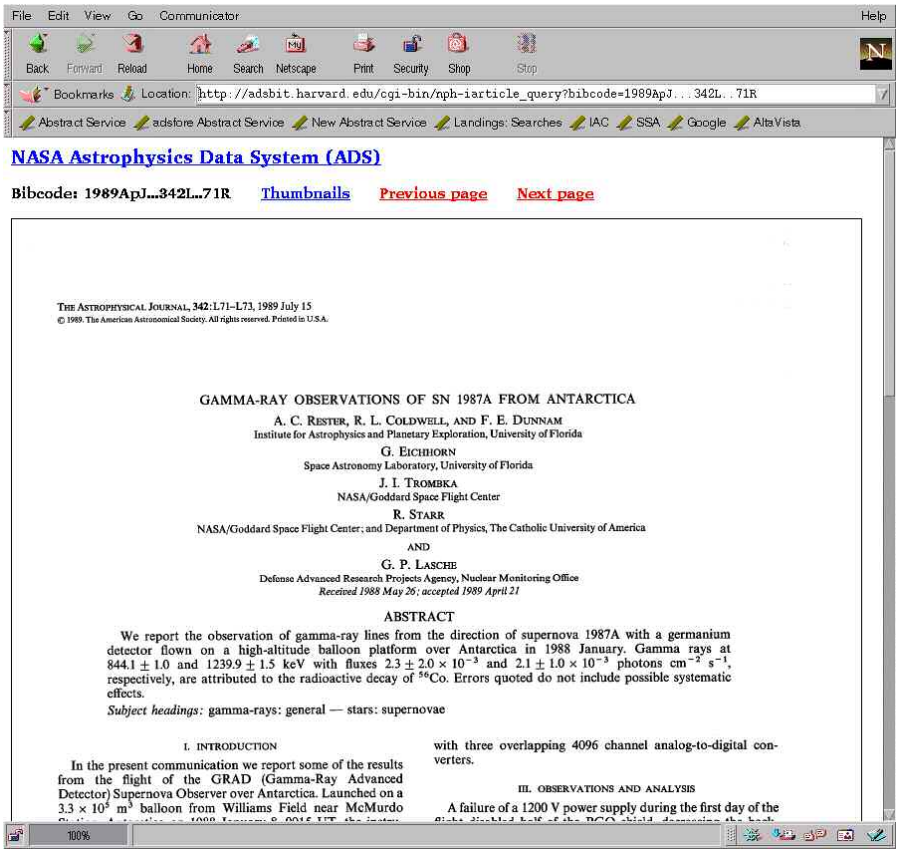

Figure 7: The article display shows a gif image of the selected journal page with the resolution selected in the user preferences.

Below the page image on the returned page are links to every page of the article individually. This allows the user to directly access any page in the article. Wherever possible, plates that have been

printed separately in the back of the journal volume have been bundled with the articles to which they belong for ease of access. The next part of the displayed document provides access to plates in that volume if the plates for this journal are separate from the articles. Another link retrieves the abstract for this article.

The next part of the page allows the printing of the article. If the browser works with HTTP persistent cookies (see section 4), there is just one print button in that section with a selection to print either the whole paper or individual pages. This print button will print the article in the format that the user has specified in the user preferences. If the browser does not handle cookies, several of the more commonly used print options are made available here.

All possible printing options can be accessed through the next link called ``More Article Retrieval Options". This page allows the user to select all possible retrieval options. These include:

i. Postscript: Access to two resolutions is provided (200 dpi and 600 dpi). For compatibility with older printers, there is also an option to retrieve Postscript Level 1 files. Postscript is a printer control language developed by Adobe (see [Adobe Postscript 1990]).

ii. PCL (Printer Control Language): This language is for printing on PCL printers such as the HP desk jets and compatibles.

iii. PDF (Portable Document Format): PDF can be viewed with the Adobe Acrobat reader ([Adobe, Inc]). From the Acrobat reader the article can be printed.

iv. TIFF (Tagged Image File Format): The original images can be retrieved for local storage. This would allow further processing like extraction of figures, or Optical Character Recognition (OCR) in order to translate the article into ASCII text.

v. Email retrieval: Articles can be retrieved through email instead of through a WWW browser. MIME (Multipurpose Internet Mail Extension, [Vaudreuil 1992]) capable email systems should be able to send the retrieved images directly to the printer, to a file, or to a viewer, depending on what retrieval option was selected by the user.

For most of the retrieval options, the data can optionally be compressed before they are sent to the user. Unix compress and GNU gzip are supported compression algorithms.

Instead of displaying the first page of an article together with the other retrieval links, the user has the option (selected through the preferences system, see section 4) to display thumbnails of all article pages simultaneously. This allows an overview of the whole article at once. One can easily find specific figures or sections within an article without having to download every page. This should be especially useful for users with slow connections to the Internet. Each thumbnail image ranges in size from only 700 bytes to 3000 bytes, depending on the user selected thumbnail image quality. The rest of this type of article page is the same as for the page-by page display option.

3.1.6 Browse Interfaces

There are several forms available to find references or articles and other relevant information. All these query forms return the short reference format as described above. One form allows access to references by specifying the journal/volume/page of the article or the bibliographic code:

http://adsabs.harvard.edu/bib_abs.html"

This form allows the user to retrieve abstracts by specifying directly a bibliographic code or the individual parts of a bibliographic (year, journal, volume, page). This can be very useful in retrieving references from article reference lists, since such reference lists generally contain enough information to use this form. The form also accepts partial codes and returns all references that match the partial code. It accepts the wildcard character `?'. The `?' wildcard stands for one character in the code. For partial codes that are shorter than 19 characters, matching is done on the first part of the bibliographic codes. For instance:

1989ApJ...341?...1

will retrieve the articles on page 1 of the ApJ (Astrophysical Journal) and ApJ Letters volume 341, regardless of the author name.

Another form allows access to the Tables of Contents (ToCs) of selected journals by month/year or volume:

http://adsabs.harvard.edu/toc_service.html

One option on that form is to retrieve the latest published issue of a particular journal. Access to the last volumes of a set of journals is also available though a page with a graphical display of selected journals' cover pages:

http://adsabs.harvard.edu/tocs.html

By clicking on a journal cover page either the last published volume of that journal or the last volume that the user has not yet read is returned, depending on the user preference settings (see section 4). The information necessary for that service is stored with the user preferences in our internal user preferences database.

A customized ToC query page is available at:

http://adsabs.harvard.edu/custom_toc.html

It will display only icons for journals that have issues available that have not been read by the user. This allows a user to see at a glance which new issues for this set of journals have been published. The set of journals that is included in the customized ToC query page can be specified in the user preferences (see section 4).

As mentioned in section 3.1.1 and in [Accomazzi et al. 2000], one important aspect of the ADS search system is the list of synonyms. Sometimes it is important for our users to be able to see what words are in a particular synonym group to properly interpret the search results. Another question that is asked is what words are in the database and how often. The list query page (linked to the words ``Authors", ``Title Words", and ``Text Words" above the corresponding entry fields on the main query form) allows the user to find synonym groups and words in the database. The user can specify either a complete word in order to find its synonyms, or a partial word with wildcard characters to find all matching words in the database. When a word without wildcard characters is specified, the list query form returns all of its synonyms (if any).

To find words matching a given pattern, the users can specify a partial word with either or both of two wildcard characters. The question mark `?' stands for any single character, the asterisk `*' stands for zero or more characters (see section 5.2.3). For a wildcard search, the list query form

returns all words in the database that match the specified pattern, together with the frequencies of these words in the database.

3.2 Direct Access Interfaces

Both abstracts and articles can be accessed directly though HTML hyperlinks. The references are identified through the bibliographic codes (or bibcodes for short) mentioned above and described in detail in [Grant et al. 2000]. The syntax for such links to access abstracts is:

http://adsabs.harvard.edu/cgi-bin/bib_query?bibcode=1989ApJ...342L..71R

Scanned articles can be accessed directly through links of the form:

http://adsabs.harvard.edu/cgi-bin/article_query?bibcode=1989ApJ...342L..71R

These links will return the abstract or scanned article respectively for the specified bibliographic code. They are guaranteed to work in this form. You may see other URLs while you use the ADS. These are internal addresses that are not guaranteed to work in the future. They may change names or parameters. Please use only links of the form described above to directly access abstracts and articles.

3.3 Embedded Queries

Embedded queries can be used to build hyperlinks that return the results of a pre-formulated query. One frequently used example is a link that returns all articles written by a specific user. The syntax for such a link is:

```
<a href="http://adsabs.harvard.edu/cgi-bin/
       abs_connect?return_req=no_params&
       param1=val1&param2=val2&...">...</a>
```

There are no spaces allowed in a URL. All blanks need to be encoded as `+'. The parameter return_req=no_params sets all the default settings. Individual settings can be changed by including the name of the specific setting and its value after the return_req=no_params parameter in the URL. A list of available parameters can be accessed at:

http://ads.harvard.edu/cgi-bin/get_field_names.pl

We try to make changes to parameters backward compatible, but that is not always possible. We encourage you to use this capability, but it is advisable to use only the more basic parameters.

This type of interface allows users to link to the ADS for a comprehensive list of references on a specific topic. Many users use this to provide an up-to-date publication list for themselves by encoding an author query into an embedded query.

3.4 Perl Script Queries

The ADS database can be used by other systems to include ADS data in documents returned from that site. This allows programmers at other sites to dynamically include the latest available information in their pages. An example is the interface that SPIE (the International Society for Optical Engineering) provides to our database. It is available at:

http://www.spie.org/incite/

This site uses Perl scripts to query our database and format the results according to their conventions. These Perl scripts are available at:

http://adsabs.harvard.edu/www/adswww-lib/

The Perl scripts allow the programmer to specify all the regular parameters. The results are returned in Perl arrays. If you use these scripts, we would appreciate it if you would credit the ADS somewhere on your pages.

3.5 Email Interface

3.5.1 Email Access to the Search System

The ADS Abstract Service can be accessed through an email interface. This access method may be especially important for users on slow or unreliable links. It will allow access to the ADS that works even if the connection is available only sometimes and/or is very slow. Access through a browser may be impossible, but email will still get through because of automatic retrying of connections.

This service is somewhat difficult to use since it involves an interface between two relatively incompatible interface paradigms. This makes describing it quite difficult as well. It is intended for users who do not have access to web browsers. If you have questions about how to use this access, please contact the ADS at ads@cfa.harvard.edu.

To get information about this capability, send email to:

adsquery@cfa.harvard.edu

with the word ``help" in the message body.

This interface accepts email messages with commands in the message body. The subject line is ignored. The commands that are available are:

1. help (see above)
2. action=URL

The second command allows a user to retrieve a document at the specified URL. Three qualifiers allow the user to specify what retrieval method to use, what format to return, and to which address to return the results:

a. method=`method'
b. return=`return-type'
c. address=`e-mail-address'

a. `method' is either `get' or `post' (without the quotes). This determines what kind of query will be executed. To retrieve a form for further queries, use the `get' method. To execute a forms query you need to know what type of query the server can handle. If you execute a forms query after retrieving the form through this service, the correct method line will already be in place. Default method is `get'.

b. `return-type' is either `text', `form', or `raw' (without the quotes). If text return is requested, only the text of the query result is returned, formatted as if viewed by a WWW browser. If form return is requested, the text of the result is returned as well as a template of the form that can again be executed with this service. If raw return is requested, the original document is returned without any processing. Default return is `form'. The capability to return MIME encoded files is in preparation.

c. `e-mail-address' specifies to which e-mail address the result should be sent. This line is optional. If no address is specified, the result is sent to the address from where the request came.

To execute a query via email, the user first retrieves the abstract query form with:

```
action=http://adsabs.harvard.edu/abstract_service.html
```

This will return an executable form. This form can be returned to the ADS in an email message to execute the query. The user enters input for the different fields as required in the forms template. For forms tags like checkboxes or radio buttons, the user can uncomment the appropriate line in the form. Comments in the form that are included with each forms field provide guidance for completing the form before submission. The text part of the form is formatted as comment lines so that the user does not have to modify any irrelevant parts of the form. The retrieval method is already set appropriately.

Alternatively, the user can send a filled out query form as retrieved from the regular query form with the "Return Query Form" button.

3.5.2 Email Access to the Scanned Articles

The scanned articles in the ADS can be retrieved via email as well. This can be either set in the user preferences (see section 4) or requested from the article display pages under ``More Article Retrieval Options".

4 Preferences

The ADS user interface is customized through the use of so-called HTTP persistent cookies (see [Kristol & Montulli 1997]). These ``cookies" are a means of identifying individual users. They are strings that are created by the server software. Web browsers that accept cookies store these identification strings locally on the user's computer. Anytime a user makes a request, the ADS software checks whether the requesting browser is capable of accepting cookies. If so, it sends a unique string to the browser and asks it to store this string as an identifier for that user. From then on, every time the same user accesses the ADS from that account, the browser sends this cookie back to the ADS server. The ADS software contains a database with a data structure for each cookie that the ADS has issued. The data structure associated with each cookie contains information such as the type of display the user prefers, whether tables should be used to format data, which mirror sites the user prefers for certain external data providers, the preferred printing format for ADS articles, and which journal volumes the user has read. It also can store the email address of the user, in case the user wants to retrieve articles via email.

The preference settings form allows the user to customize these settings. It includes a field for the user name as well as the email address. However, neither is necessary for the functioning of any of the features of the ADS. The system is completely anonymous. None of the information stored through this cookie system is made available to anybody outside the ADS. There is no open

interface to this database and the database files are not accessible through the WWW. Any particular user can only access their own preferences, not the preferences set by any other user.

Most of these preferences can be set by the user through a WWW forms interface. Some fields in a user's preference record are for ADS internal use only. For instance the system is being used to display announcements to users in a pop-up window. The cookie database remembers when the message was last displayed, so that each message is displayed only once to each user.

This preference saving system also allows each user to store a query form with filled-in fields. This enables users to quickly query the ADS for a frequently used set of criteria.

Another possibility for customizing forms is to fill out a form, click on ``Return Query Form'', and then to bookmark that returned form. Whenever you go to this bookmark, you will get the filled out form.

The cookie identification system is implemented as a Berkeley DBM (DataBase Manager) database with the cookie as the record key. The data block that is stored in the database is a C structure. The binary settings (e.g. ``Use Tables'', or ``Use graphical ToC Page'') are stored as bits in several preference bytes. Other preferences are stored as bytes, short integer, or long integer, depending on their dynamic range.

5 Search Engine

5.1 General

The basic design assumption behind the search engine, and other user interfaces, is that the user is an expert astronomer. This differs from the majority of information retrieval systems, which assume that the user is a librarian. The default behavior of the system is to return more relevant information, rather than just the most relevant information, assuming that the user can easily separate the wheat from the chaff. In the language of information retrieval this is favoring recall over precision.

The search engine software is written in C. It accepts as input a structure that contains all the search fields and flags for the user specified settings and filters. For each search field that contains user input a separate POSIX (Portable Operating System Interface) thread is started that searches the database for the terms specified in that field. The results obtained for each term in that field are combined in the thread according to the specified combination logic. The resulting list of references is returned to the main thread. The main thread combines the results from the different field searches and calculates the final score for each reference. The final combined list with the scores is returned to the user interface routines that format the results according to the user specified output format.

5.2 Searching

5.2.1 Database Files

The abstracts are indexed in separate fields: Author names, titles, abstract text, and objects. Each of these fields is indexed similarly into an index file and a list file (see [Accomazzi et al. 2000]). The index file contains a list of all terms in the field together with the frequency of the term in the database, and the position and length of two blocks of data in the list file. One block contains all

references that include the exact word as specified. The other block contains all references that contain either the word or any of its synonyms.

A search for a particular word in the index file is done through a binary search in the index. The indexes are resident in memory, loaded during boot time (see section 5.6). Once the word is found in the index, the position and length of the data block is used to directly access the data block in the list file. This data block contains the identifier for each reference that contains the search word.

5.2.2 Synonym Searches

As mentioned above, the index files contain information about two blocks of data, the data for the individual word and for the synonym group to which this word belongs. When a search with synonym lookup enabled is requested, the block of data for the whole synonym group is used, otherwise the data for only the individual word is returned. All the processing that enables these two types of searches is done during indexing time, therefore the speeds for both searches are similar.

Even though our synonym list is quite extensive (see [Accomazzi et al. 2000]) our users will sometimes use words that are not in the database or the synonym list. In these cases the search software uses a stemming algorithm from the Unix utility ispell to find the stem of the search word and the searches for the word stem. The indexing software has indexed the stems of all words in the database together with their original words (see [Accomazzi et al. 2000]). This word stemming is done as a last resort if no regular match has been found in an attempt to find any relevant references.

5.2.3 Wildcard Searches

In order to be able to search for families of words, a limited wildcard capability is available. Two wildcard characters are defined: The question mark `?' is used to specify a single wildcard character and the asterisk `*' is used to specify zero or more wildcard characters. The `?' can be used anywhere in a word. For instance a search for *M1?* will find all Messier objects between M10 and M19. A search for *a?sorb* will find references with *absorb* as well as *adsorb*.

The asterisk can only be used at the beginning or at the end of a word. For instance *3C** searches for all 3C objects. **sorb* searches for words that end in *sorb* like absorb, desorb, etc. When synonym replacement is on, all their synonyms (e.g. absorption) will be found as well. The `?' and the `*' can be combined in the same search string.

5.3 Results Combining within a Field

5.3.1 Combining results

As mentioned above, the user can select between four types of combination methods: ``OR'', ``AND'', simple logic, and full boolean logic. For the first three cases, the references for all search terms are retrieved and sorted first. The reference lists are then merged by going through the sorted reference lists sequentially and synchronously and selecting references according to the chosen logic.

The search algorithm for the full boolean logic is different. The boolean query is parsed from left to right. For each search term a function is called that finds the references for this term. A search term is either a search word, a phrase, or an expression enclosed in parentheses. If the search term contains other terms (if it is enclosed in parentheses), the parsing function is called recursively.

The next step is to determine the boolean operator that follows the search term, and then to evaluate the next search term after the operator. Once the reference lists for the two terms and the combining operator are determined, the two lists are combined according to the operator. This new list is then used as the first term of the next expression.

If the boolean operator is `OR', the combining of the lists is deferred, and the next operator and search term are evaluated. This ensures the correct precedence of `OR' and `AND' operators.

The `NOT' operator is implemented by getting the reference list for the term, making a copy of all references in the database, and then removing the references from the search term from the complete list. This yields a very large list of references. If the first search term in a boolean expression is a `NOT' term, the search will take very long, because this large list has to be propagated through all the subsequent parsing of the boolean expression. Care should therefore be taken to put a `NOT' term to the right of at least one other term, since processing is done left to right.

5.3.2 Scoring

In addition to the information about the references for each word, the index file also contains its frequency in the database. The frequency is already pre-calculated as int(10000/(log(frequency))) during indexing (see [Accomazzi et al. 2000]). This saves considerable time during execution of the search engine since all server calculations can be done as integer computations, no floating point operations are necessary. During the first part of the search, this frequency is attached to each retrieved reference. In the next step, the retrieved references are combined according to the selected combination logic for that field.

For `OR' combination logic, the lists retrieved for each word are merged and uniqued. As described in section 5.6, this is done by going through the sorted reference lists synchronously and adding each new reference to the output list. The score for that reference is determined by adding up the frequencies from each of the lists for weighted scoring, or by setting a score equal to the number of matched words for proportional scoring.

For `AND' combination logic, only references that appear in every one of the lists are selected. Each of these references receives a score of 1.

For simple logic and full boolean logic, the score for the returned references is determined only from the search terms that were combined with `OR'. All words that have mandatory selection criteria (prefixed by `+' or `-' in simple logic, and combined with `AND' or `NOT' in full boolean logic) do not affect the final score.

5.4 Combining Results among Fields

5.4.1 Combining

After the POSIX threads for each search field are started, the main program waits for all threads to complete the search. When all searches are completed the search engine combines the results of the different searches according to the selected settings. If for instance one field was selected as required for selection in the settings section of the query form, only references that were found in the search for that field will be in the final reference list. The combined list is then uniqued and sorted by score. The resulting list of references is passed back to the user interface software.

If the user did not specify any search terms, a date range has to be selected. The software queries the database for all references in the selected date range and uses this list for further processing, e.g. filtering (see section 5.5).

5.4.2 Scoring

The score for each reference in the final results list is determined by adding the scores from each list multiplied by the user specified weight for each field and then normalizing the score such that a reference that matches all search terms from all fields receives a score of 1.

5.5 Selection Filters

After the search is completed according to the specified search words and the settings that control the combination logic and the scoring algorithms, the resulting list of references can be filtered according to several criteria. During the design of the software a decision had to be made whether to filter the results while selecting the references or after completing the search. The first approach has the advantage that the combining of the selected references will be faster because fewer references need to be combined. The second approach has the advantage that the first selection is faster. We chose the second approach because, except for selecting by publication date, only a small number of queries use filtering (see section 8). Because of that, filtering by publication date is done during selection of the references, while the other filtering is done after the search is completed.

References can be filtered by five criteria:

1. Entry date in the data base
2. Minimum score
3. Journal
4. Available links
5. Group membership

1. + 2. Entry date and minimum score. These two filters can be used to query the database automatically on a regular basis for new information that is relevant to a selected topic. The user can build a query form that returns relevant references, and then save this query form locally. This query form can then be sent to the ADS email interface (see above) on a weekly or monthly basis. By specifying an entry day of -7 for instance, the query will retrieve all references that fit the query and that were entered within the last seven days. The minimum score can be used to limit the returned number of references to only the ones that are really relevant. The references are returned via email as described in section 3.5 about the email interface.

3. Journal filter. This filter allows the user to select references from individual journals or groups of journals. Available options for this filter are:

a. All journals (default)
b. Refereed journals only
c. Non-refereed journals only
d. Selected journals

If the last option is selected, the user can specify one or more journal abbreviations (e.g. ApJ, AJ (Astronomical Journal)) that should be selected. More than one abbreviation can be specified by separating them with semicolons or blanks. The filter for journals can also include the volume number (but not the page number). The journal abbreviation is compared with the bibliographic

codes over the length of the specified abbreviation. For instance if the user specifies *ApJ*, the system selects all articles published in the ApJ, ApJ Supplement and ApJ Letters. *ApJ.*. will select only articles from the ApJ and the ApJ Letters. A special abbreviation, *ApJL* will select only articles from the ApJ Letters. If a journal abbreviation is specified with a prepended `-', all references that are NOT from that journal are returned. The journal abbreviations (or bibstems) used in the ADS are available at:

http://adsdoc.harvard.edu/abs_doc/journal_abbr.html

4. Available links. This filter allows the user to select references that have specific other information available. The returned references can be filtered for instance to include only references that have data links or scanned articles available. As an example, a user needs to find on-line data about a particular object. A search for that object in the object field and a filter for references with on-line data returns all articles about that object that link to on-line data.

5. Groups. We provide the capability to build a reference collection for a specific subset of references. This can be either articles written by members of a particular research institute or about a particular subject. Currently there are 8 groups in the ADS. We encourage larger institutes or groups to compile a list of their references and send it to us to be included in the list of groups.

5.6 Optimizations

The search engine is entirely custom-built software. As mentioned in the introduction, the first version of the Abstract Service used commercial database software. Because of too many restrictions and serious performance problems, a custom-designed system was developed. The main design goal was to make the search engine as fast as possible. The most important feature that helped speed up the system was the use of permanent shared memory segments for the search index tables. In order to make searching fast, these index tables need to be in Random Access Memory. Since they are tens of megabytes long, they cannot be loaded for each search. The use of permanent shared memory segments allows the system to have all the index tables in memory all the time. They are loaded during system boot. When a search engine is started, it attaches to the shared segments and has the data available immediately without any loading delays. The shared segments are attached as read-only, so even if the search engine has serious bugs, it cannot compromise the integrity of the shared segments. Using shared segments with the custom-built software improved the speed of a search by a factor of 2 - 20, depending on the type of search.

Access to the list files (see section 5.2) was optimized too. These files cannot be loaded into memory since they are too large (each is over one hundred megabytes in size). To optimize access to these files, they are memory mapped when they are accessed for the first time. From then on they can be accessed as if they were arrays in memory. The data blocks specified in the index tables can be accessed directly. Access is still from file, but it is handled through the paging in the operating system, which is much more efficient, rather than through the regular I/O system.

Once the search engine was completed and worked as designed, it was further optimized by profiling the complete search engine and then optimizing the modules that used significant amounts of time. Further analysis of the performance of the search engine revealed instances where operations were done for each search that could be done during indexing of the data and during loading of the shared segments. Overall these optimizations resulted in speed improvements of a factor of more than 10 over the performance of the first custom-built version. These optimizations were crucial for the acceptance of the ADS search system by the users.

In order to further speed up the execution, the search engine uses POSIX threads to exploit the inherent parallel nature of the search task. The search for each field, and in the case of the object field for each database, is handled by a separate POSIX thread. These threads execute in parallel, which can provide speedups in our multiprocessor server. Even for single processor systems this will provide a decrease in search time, since each thread sometimes during its execution needs to wait for I/O to complete. During these times other threads can execute and therefore decrease the overall execution time of a search.

Another important part of the optimization was the decision on how to structure the index and list files. The index files contain the word frequency information that is used to calculate scores for the weighted scoring (see section 3.1.3). The score for a matching reference is calculated from the inverse logarithm of the frequency of the word in the database. This requires time consuming floating point calculations. To avoid these calculations during the searches, the floating point arithmetic is done at indexing time. The index file contains the inverse log of the word frequency multiplied by a normalization factor of 10,000. This allows all subsequent calculations to be done in integer arithmetic, which is considerably faster than floating point calculations.

Overall, these optimizations improved the speed of the searches by two orders of magnitude between the original design using a commercial database and the current software.

6 Database Mirroring

All of the software development and data processing in the ADS has been carried out over the last 6 years in a UNIX environment. During the life of the project, the work group-class server used to host the ADS services has been upgraded three times to meet the increasing use of the system. The original dual processor Sun 4/690 used at the inception of the project was replaced by a SparcServer 1000E with two 85MHz Supersparc CPU modules in 1995 and subsequently an Ultra Enterprise 450 with two 300MHz Ultrasparc CPUs was purchased in 1997. In 2001 two Sun Sparc Ultra-80 servers were acquired to run the search system and the indexing and mirroring system. The article service is now hosted by the Ultra Enterprise 450 server.

Soon after after the inception of the article service in 1995 it became clear that for most ADS users the limiting factor when retrieving data from our computers was bandwidth rather than raw processing power. With the creation of the first mirror site hosted by the CDS in late 1996, users in different parts of the world started being able to select the most convenient database server when using the ADS services, making best use of bandwidth available to them. At the time of this writing, there are nine mirror sites located on four different continents, and more institutions have already expressed interest in hosting additional sites. The administration of the increasing number of mirror sites requires a scalable set of software tools which can be used by the ADS staff to replicate and update the ADS services both in an interactive and in an unsupervised fashion.

The cloning of our databases on remote sites has presented new challenges to the ADS project, imposing additional constraints on the organization and operation of our system. In order to make it possible to replicate a complex database system elsewhere, the database management system and the underlying data sets have to be independent of the local file structure, operating system, and hardware architecture. Additionally, networked services which rely on links with both internal and external web resources (possibly available on different mirror sites) need to be capable of deciding how the links should be created, giving users the option to review and modify the system's linking strategy. Finally, a reliable and efficient mechanism should be in place to allow unsupervised database updates, especially for those applications involving the publication of time-critical data.

The next sections describe the implementation of an efficient model for the replication of our databases to the ADS mirror sites. In section 6.1 we describe how system independence has been achieved through the parameterization of site-specific variables and the use of portable software tools. In section 6.2 we describe the approach we followed in abstracting the availability of network resources through the implementation of user-selectable preferences and the definition of site-specific default values. In section 6.3 we describe in more detail the paradigm used to implement the synchronization of different parts of the ADS databases.

6.1 System Independence

The database management software and the search engine used by the ADS bibliographic services have been written to be independent from system-specific attributes to provide maximum flexibility in the choice of hardware and software in use on different mirror sites. We currently support the Sparc/Solaris and x86/Linux servers. Given the current trends in hardware and operating systems, we expect to standardize to GNU/Linux systems in the future.

Hardware independence was made possible by writing portable software that can be either compiled under a standard compiler and environment framework (e.g. the GNU programming tools, [Loukides & Oram 1996]) or interpreted by a standard language (e.g. PERL version 5, [Wall, Christiansen & Schwartz 1996]). Under this scheme, the software used by the ADS mirrors is first compiled from a common source tree for the different hardware platforms on the main ADS server, and then the appropriate binary distributions are mirrored to the remote sites.

One aspect of our databases which is affected by the specific server hardware is the use of binary data in the list files, since binary integer representations depend on the native byte ordering supported by the hardware. With the introduction of a mirror site running Digital UNIX in the summer of 1999, we were faced with having to decide whether it was better to start maintaining two versions of the binary data files used in our indices or if the two integer implementations should be handled in software. While we have chosen to perform the integer conversion in software for the time being given the adequate speed of the hardware in use, we may revisit the issue if the number of mirror sites with different byte ordering increases with time.

Operating System independence is achieved by using a standard set of public domain tools abiding to well-defined POSIX standards ([IEEE 1995]). Any additional enhancements to the standard software tools provided by the local operating system is achieved by cloning more advanced software utilities (e.g. the GNU shell-utils package) and using them as necessary. Specific operating system settings which control kernel parameters are modified when appropriate to increase system performance and/or compatibility among different operating systems (e.g. the parameters controlling access to the system's shared memory). This is usually an operation that needs to be done only once when a new mirror site is configured.

File-system independence is made possible by organizing the data files for a specific database under a single directory tree, and creating configuration files with parameters pointing to the location of these top-level directories. Similarly, host name independence is achieved by storing the host names of ADS servers in a set of configuration files.

6.2 Site Independence

While the creation of the ADS mirror sites makes it virtually impossible for users to notice any difference when accessing the bibliographic databases on different sites, the network topology of a

mirror site and its connectivity with the rest of the Internet play an important role in the way external resources are linked to and from the ADS services. With the proliferation of mirror sites for several networked services in the field of astronomy and electronic publishing, the capability to create hyperlinks to resources external to the ADS based on the individual user's network connectivity has become an important issue.

The strategy used to generate links to networked services external to the ADS which are available on more than one site follows a two-tiered approach. First, a ``default'' mirror can be specified in a configuration file by the ADS administrator. The configuration file defines a set of parameters used to compose URLs for different classes of resources, lists all the possible values that these parameters may assume, and then defines a default value for each parameter. Since these configuration files are site-specific, the appropriate defaults can be chosen for each of the ADS mirror sites depending on their location. ADS users are then allowed to override these defaults by using the ``Preference Settings'' system (see section 4) to select any of the resources listed under a category as their default one. Their selection is stored in a site-specific user preference database which uses an HTTP cookie as an ID correlating users with their preferences.

In order to create links to external resources which are a function of a user's preferences, we store the parameterized version of their URLs in the property databases. The search engine expands the parameter when the resource is requested by a user according to the user's preferences. For instance, the parameterized URL for the electronic paper associated with the bibliographic entry `1997ApJ...486...42G` can be expressed as `$UCP$/cgi-bin/resolve?1997ApJ...486...42G`. Assuming the user has selected the first entry as the default server for this resource, the search engine will expand the URL to the expression:

`http://www.journals.uchicago.edu/cgi-bin/resolve?1997ApJ...486...42G`
This effectively allows us to implement simple name resolution for a variety of resources that we link to. By saving these settings in a user preference database indexed on the user HTTP cookie ID (see section 4), users only need to define their preferences once and our interface will retrieve and use the appropriate settings as necessary.

6.3 Mirroring Software

The software used to perform the actual mirroring of the databases consists of a main program running on the ADS master site initiating the mirroring procedure, and a number of scripts, run on the mirror sites, which perform the transfer of files and software necessary to update the database. The paradigm we adopted in creating the tools used to maintain the mirror sites in sync is based on a ``push'' approach: updates are always started on the ADS main site. This allows mirroring to be easily controlled by the ADS administrator and enables us to implement event-triggered updating of the databases. The main mirroring program, which can be run either from the command line or through the Common Gateway Interface (CGI), is a script that initiates a remote shell session on the remote sites to be updated, sets up the environment by evaluating the mirror sites' and master site's configuration files, and then runs scripts on the remote sites that synchronize the local datasets with the ADS main site.

The updating procedures are specialized scripts which check and update different parts of the database and database management software (including the procedures themselves). For each component of the database that needs to be updated, synchronization takes place in two steps, namely the remote updating of files which have changed to a staging directory, and the action of making these new files operational. This separation of mirroring procedures has allowed us to enforce the proper checks on integrity and consistency of a data set before it is made operational.

The actual comparison and data transfer for each of the files to be updated is done by using a public domain implementation of the rsync algorithm ([Tridgell 1999]). The advantages of using rsync to update data files rather than using more traditional data replication packages are summarized below.

1) Incremental updates: rsync updates individual files by scanning their contents, computing and comparing checksums on blocks of data within them, and copying across the network only those blocks that differ. Since during our updates only a small part of the data files actually changes, this has proven to be a great advantage. Recent implementations of the rsync algorithms also allow partial transfer of files, which we found useful when transferring the large index files used by the search engine. In case the network connection is lost or times out while a large file is transferred, the partial file is kept on the receiving side so that transfer of additional chunks of that file can continue where it left off on the next invocation of rsync.

2) Data integrity: rsync provides several options that can be used to decide whether a file needs updating without having to compare its contents byte by byte. The default behavior is to initiate a block by block comparison only if there is a difference in the basic file attributes (time stamp and file size). The program however can be forced to perform a file integrity check by also requesting a match on the 128-bit MD4 checksum for the files.

3) Data compression: rsync supports internal compression of the data stream sent between the master and mirror hosts by using the zlib library ([Deutsch & Gailly 1996]).

4) Encryption and authentication: rsync can be used in conjunction with the Secure Shell package ([Ylonen et al. 1999]) to enforce authentication between rsync client and server host and to transfer the data in an encrypted way for added security. Unfortunately, since all of the ADS mirror sites are outside of the U.S., transfer of encrypted data could not be performed at this time due to restrictions and regulations on the use of encryption technology.

5) Access control: the use of rsync allows the remote mirror sites to retrieve data from the master ADS site using the so-called anonymous rsync protocol. This allows the master site to exercise significant control over which hosts are allowed to access the rsync server, what datasets can be mirrored, and does not require remote shell access to the main ADS site, which has always been the source of great security problems.

During a typical weekly update of the ADS astronomy database, as many as 1% of the text files may be added or updated, while the index files are completely recreated. By checking the attributes of the individual files and transferring only the ones for which either time stamp or size has changed, the actual data which gets transferred when updating the collection of text files is of the order of 1.7% of the total file size (12MB vs. 700MB). By using the incremental update features of rsync when mirroring a new set of index files, the total amount of data being transferred is of the order of 38% (250MB vs. 650MB).

6.4 Mirror Sites

The ADS is mirrored world-wide at 10 sites. Table 2 shows the current mirror sites and their URLs.

Table 2: ADS Mirror Sites.

USA	Harvard-Smithsonian Center for Astrophysics, Cambridge, MA	http://ads.harvard.edu/
France	Centre des Données astronomiques de Strasbourg	http://cdsads.u-strasbg.fr/

Japan	National Astronomical Observatory, Tokyo	http://ads.nao.ac.jp/
Chile	Pontificia Universidad Católica, Santiago	http://ads.astro.puc.cl/
Germany	European Southern Observatory, Garching	http://esoads.eso.org/
Great Britain	University of Nottingham, Nottingham	http://ukads.nottingham.ac.uk/
China	Beijing Astronomical Observatory, Beijing	http://baoads.bao.ac.cn/
India	Inter-University Centre for Astronomy and Astrophysics, Pune	http://ads.iucaa.ernet.in/
Russia	Institute of Astronomy, Russian Academy of Science, Moscow	http://ads.inasan.rssi.ru/
Brazil	Observatorio Nacional, Rio de Janeiro	http://ads.on.br/

Setting up a mirror site is fairly easy. The hosting institution has to provide a server and an Internet connection. Such an abstract mirror site can now run on a Linux PC with 20 Gb of disk space. A partial article mirror site can run on as little as 80 Gb of disk space. If you are interested in having a mirror site, please contact Dr. Guenther Eichhorn at gei@cfa.harvard.edu for detailed requirements.

7 Query Examples

7.1 Examples

The ADS answers over 8,000,000 queries per year, covering a wide range of possible query type, from the simplest (and most popular): ``give me all the papers written by (some author),'' to complex combinations of natural language described subject matter and bibliometric information. Each query is essentially the sum of simultaneous queries (e.g. an author query and a title query), where the evidence is combined to give a final relevance ranking (e.g. [Belkin, et. al 1995]).

Here we show examples of simple, but sophisticated queries, to give an indication of what is possible using the system. A detailed description of available query options is in section 3. We encourage the reader to perform these queries now, to see how the passage of time has changed the results.

Figure 1 shows how to make the query ``what papers are about the metallicity of M87 globular clusters?'' This was the first joint query made after the SIMBAD-ADS connection was completed in 1993.

There are 1,914 papers on M87 in SIMBAD, NED, or both; there are 8,546 papers which contain the phrase ``globular cluster'' in ADS, and there are 33,896 papers in ADS containing ``metallicity'' or a synonym (abundance is an example of a synonym for metallicity). The result, which comes in a couple of seconds, is a list of just those 89 papers desired.

Five different indices are mixed in this query: the SIMBAD object--bibcode index query on M87 is logically OR'd with the NED object--refcode index query for M87. The ADS phrase index query for ``globular cluster'' is (following the user's request) logically AND'd with the ADS word index query on metallicity, where metallicity is replaced by its group of synonyms from the ADS astronomy synonym list (this replacement is under user control). If the user requires a perfect match, then the

combination of these simultaneous queries yields the list of 89 papers shown in figure 4. Before the establishment of the Urania core queries like this were nearly impossible.

Another simple, but very powerful method for making ADS queries is to use the ``Find Similar Abstracts'' feature. Essentially this is an extension of the ability to make natural language queries, whereby the user can choose one or more abstracts to become the natural language query. This can be especially useful when one wants to read in depth on a subject, but only knows one or two authors or papers in the field. This is a typical situation for many researchers, but especially for students.

As an example, suppose one is interested in the first author's (1974) PhD thesis work. Making an author query on ``Eichhorn, G'' gets a list of his papers, including his thesis. Next one calls up the abstract of the thesis, goes to the bottom of the page, where the ``Find Similar Abstracts'' feature is found, and clicks the ``Send'' button. Alternatively, such feedback queries can be executed from the bottom of the first results list. Figure 8 shows the top of the list returned as a result. These are papers listed in order of similarity to the first author's (1974) thesis; note that the thesis itself is on top with a score of 1.0, as it matches itself perfectly. This list is a detailed subject matter selected custom bibliography.

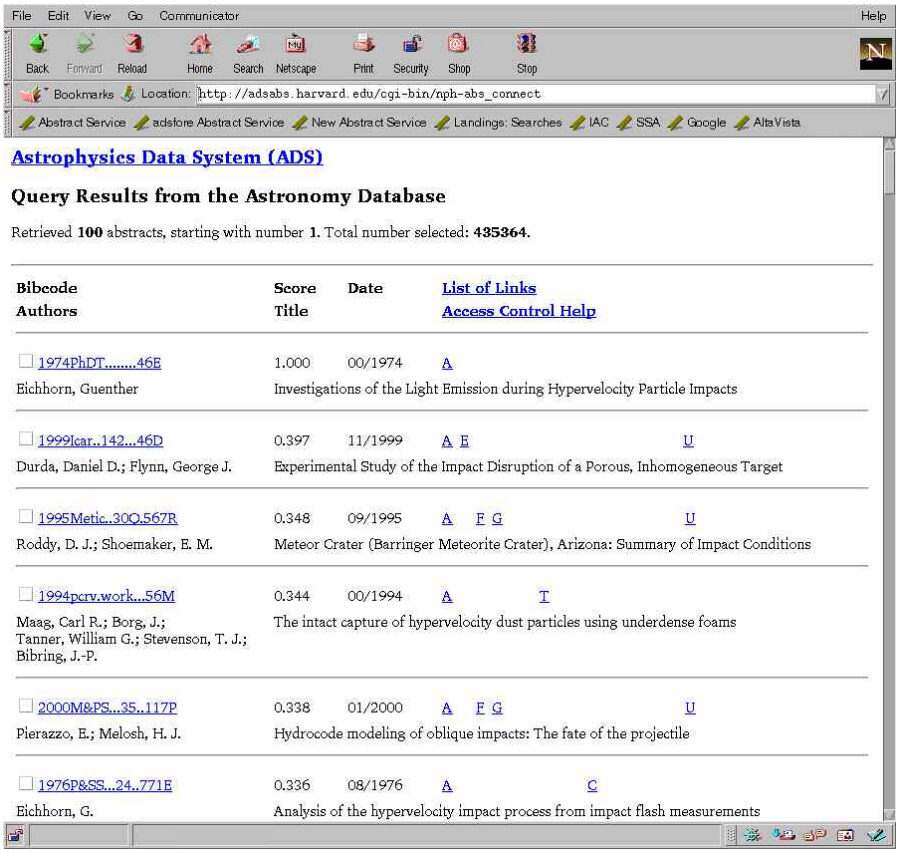

Figure 8: The top of the list of papers returned when Guenther Eichhorn's (1974) thesis is used as the query.

8 Use of the System

The ADS is used by a large majority of professional astronomers world-wide on a daily basis, as well as by many other researchers and non-scientists. This section shows some of the access statistics of the ADS.

The usage of the ADS has bee continuously increasing since its start in 1993. Figure 9 shows the number of references retrieved per month for the life of the ADS.

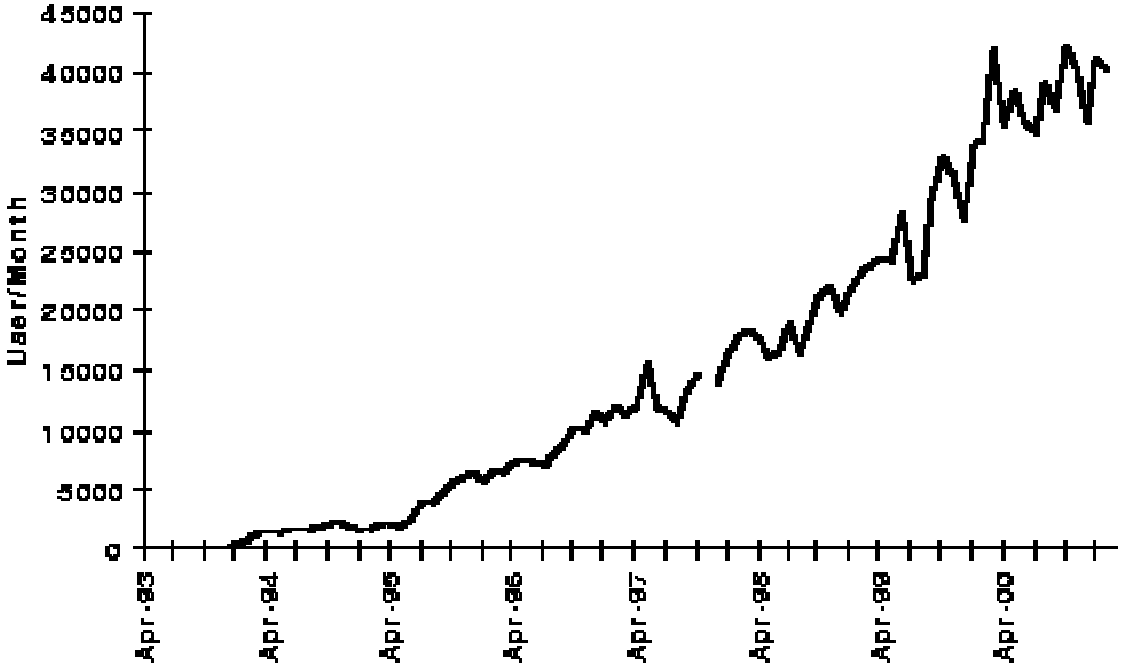

Figure 9: Number of references retrieved per month.

In March, 2001 50,000 users made 863,000 queries, and received 28,000,000 bibliographic references, 230,000 full text articles and 613,000 abstracts, as well as citation histories, links to data, and links to other data centers. Of the 230,000 full-text articles accessed through the ADS □ 50% were via pointers to the electronic journals.

ADS users access and print (either to the screen, or to paper) more actual pages than are printed in the press runs of any journal in astronomy. In March, 2001, 1.1 million page images were downloaded from the ADS archive of scanned bitmaps. About 75% of these were sent directly to a printer, 22% were viewed on the computer screen, and 2% were downloaded into files; viewing thumbnail images make up the rest.

The ADS is used 24 hours per day. The distribution of queries throughout the day is shown in figure 10. This figure shows the number of queries at our different mirror sites. The usage distribution data are for the time period from 1 November 2000 to 31 March 2001, not the full year, to avoid complications due to different periods where daylight savings time is in effect. The USA distribution is shown in two parts, one for requests coming from US host (SAO-US), the other for requests coming from non-US hosts (SAO-Non-US).

Most of the individual curves show a distinct two-peaked basic shape, with additional smaller peaks in some cases. This distribution of queries over the day shows the usage throughout a workday, with a small minimum during lunch hour. The Distribution of accesses to the US site from US hosts is not quite the same, probably because the US covers 3 time zones.

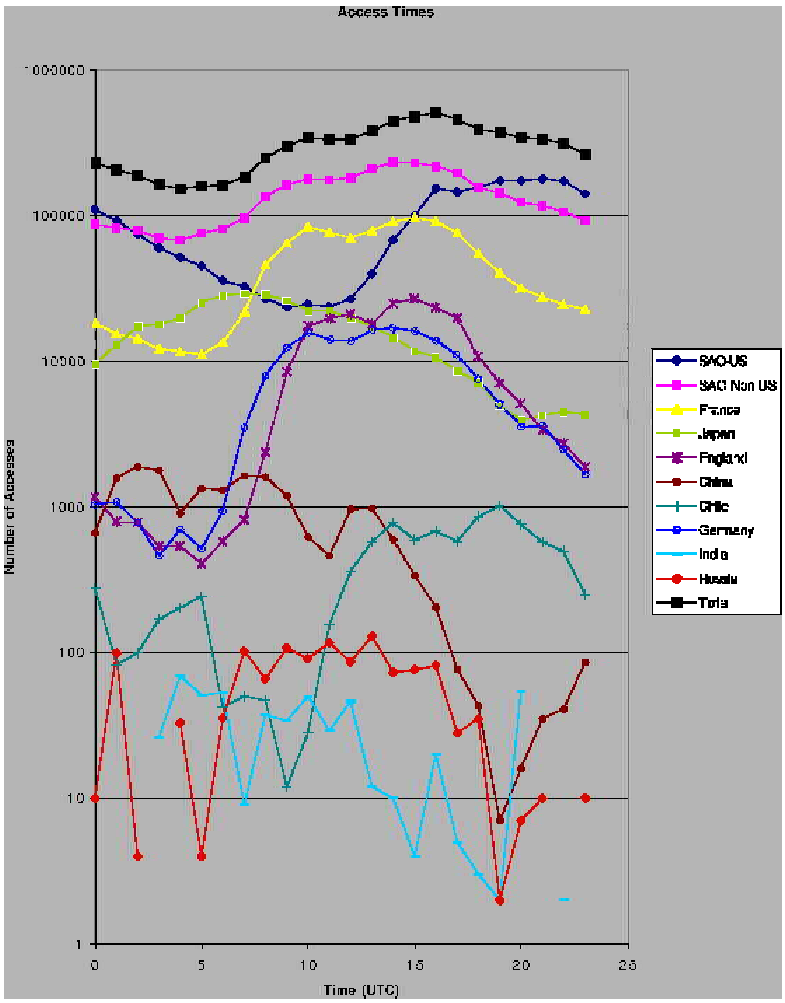

Figure 10: Number of queries per hour as a function of the time of day

The distribution for Germany and England are very similar, with the English distribution shifted by 1 hour, as is to be expected because of the time difference. The distribution of accesses to the French mirror site is broader than the German or English distribution, but with the peaks in about the same place. The broader distribution is probably due to the fact that the French mirror site is the oldest, and therefore is used by more people world-wide, which tends to wash out the distinct time dependence.

The shape of the accesses to the ADS mirror in France is the same as the shape of the non-US access to the SAO site. This indicates that the large majority of the non-US use on the SAO site is from European users. This non-US usage at the US site is about three times as high as the total usage of the ADS mirror site at the CDS in France. The reason for this is most probably the fact that the connectivity within Europe is sometimes not yet very good. We know that for instance that our users in England and Sweden have better access to the main ADS site in the USA than to our mirror site in France. The same is true for other parts of Europe.

Another reason for the use of the USA site by European users is the fact that our European mirror sites do not yet have the complete set of scanned articles on-line. This forces some users to access the main ADS site in order to retrieve scanned articles.

There is a slight peak in the distribution of queries to the NAO mirror in Japan around 21:00 UTC (Universal Time Coordinated, formerly Greenwich Mean Time). This is probably due to US west

coast users using the Japanese mirror site instead of the US site. The access to Japan is frequently very fast and response times from Japan may be better than from SAO during peak traffic times.

The distribution of accesses to China, as expected, is somewhat similar to the distribution of accesses to Japan. The access to Chile peaks about 5 hours after the accesses to Europe as expected. The statistics for accesses to the mirror sites in India and Russia are not good enough to allow any comparison.

In total, the usage from US hosts is about 1/3 of the total usage of the ADS, 2/3 is from non-US sites, mostly from Europe.

9 Conclusion

The ADS provides free access to most of the astronomical literature. It has profoundly changed the way astronomers do their research. We hope that it will continue to facilitate astronomical research in particular in countries that do not have easy access to libraries with astronomical literature. It should also allow new studies of the historical literature that are so far very difficult or impossible. We welcome any questions and suggestions on how to improve the ADS services. Please contact us at

ads@cfa.harvard.edu

10 Acknowledgment

Funding for this project has been provided by NASA under NASA Grant NCC5-189.

11 References

Accomazzi et al. 2000
Accomazzi, A., Eichhorn, G., Grant, C. S., Kurtz, M. J., & Murray, S. S. 2000, ``The NASA Astrophysics Data System: Architecture," A&AS, 143, 85

Adobe, Inc
Adobe Acrobat Reader, http://www.adobe.com/prodindex/acrobat/alternate.html

Adobe Postscript 1990
Adobe Systems 1990, ``Postscript Language Reference Manual, Second Edition," Addison-Wesley, Reading, MA

Bardeen et al. 1986
Bardeen, J. M., Bond, J. R., Kaiser, N., & Szalay, A. S. 1986, ``The statistics of peaks of Gaussian random fields," ApJ, 304, 15

Belkin, et. al 1995
Belkin, N.J., Kantor, P., Fox, E.A., and Shaw, J.A. 1995, ``Combining the Evidence of Multiple Query Representations for Information Retrieval," Information Processing and Management 31, 431.

Boyce 1995
Boyce, P. B. 1995, ``The Electronic ApJ Letters," American Astronomical Society Meeting, 187, 3801

Boyce 1996
Boyce, P. 1996, ``Journals, Data and Abstracts Make an Integrated Electronic Resource," American Astronomical Society Meeting, 189, 0603

Bromley 1994

Bromley, B. C. 1994, ``Large-scale structure of the universe: a clustering analysis,'' Ph.D. Thesis, Dartmouth College

CIA 1999
CIA World Factbook, 1999, US Government Publications, http://www.cia.gov/cia/publications/factbook/

Corbin & Coletti 1995
Corbin, B. G.& Coletti, D. J. 1995, ``Digitization of Historical Astronomical Literature,'' Vistas in Astronomy, 39, 161

Davis & Peebles 1983
Davis, M. & Peebles, P. J. E. 1983, ``A survey of galaxy redshifts. V - The two-point position and velocity correlations, ``ApJ, 267, 465

Demleitner, et al. 1999
Demleitner, M., Accomazzi, A., Eichhorn, G., Grant, C.S., Kurtz, M.J.,& Murray, S.S., 1999, ``Looking at 3,000,000 References Without Growing Grey Hair,'' American Astronomical Society Meeting, 195, 8209

Deutsch & Gailly 1996
Deutsch, L. P. & Gailly, J. 1996, ``ZLIB Compressed Data Format Specification, version 3.3,'' RFC 1950, Internet Engineering Task Force

Egret et al. 1991
Egret, D, Wenger, M., & Dubois, P. 1991, ``The SIMBAD Astronomical Database,'' ``Databases & On-line Data in Astronomy,'' D. Egret & M. Albrecht, Eds, Kluwer Acad. Publ., 79

Eichhorn 1994a
Eichhorn, G. 1994, ``An Overview of the Astrophysics Data System,'' Experimental Astronomy, 5, 205

Eichhorn et al. 1994b
Eichhorn, G., Kurtz, M. J., Accomazzi, A., Grant, C. S.,& Murray, S. S. 1994, ``Full Journal Articles in the ADS Astrophysics Science Information and Abstract Service,'' American Astronomical Society Meeting, 185, 4104

Eichhorn et al. 1995a
Eichhorn, G., Accomazzi, A., Grant, C. S., Kurtz, M. J. & Murray, S. S. 1995, ``Access to the Astrophysics Science Information and Abstract System,'' Vistas in Astronomy, 39, 217

Eichhorn et al. 1995b
Eichhorn, G., Murray, S. S., Kurtz, M. J., Accomazzi, A. & Grant, C. S. 1995, ``The New Astrophysics Data System,'' ASP Conf. Ser. 77: Astronomical Data Analysis Software and Systems IV, 28

Eichhorn 1996a
Eichhorn, G. 1996, ``The Virtual Library,'' Sky & Telescope, 92, 81

Eichhorn et al. 1996b
Eichhorn, G., Accomazzi, A., Grant, C. S., Kurtz, M. J. & Murray, S. S. 1996, ``Various Access Methods to the Abstracts in the Astrophysics Data System,'' ASP Conf. Ser. 101: Astronomical Data Analysis Software and Systems V, 569

Eichhorn 1997a
Eichhorn, G. 1997, ``The digital library of the Astrophysics Data System,'' Astrophys. Space Sci., 247, 189

Eichhorn et al. 1997b
Eichhorn, G., Kurtz, M. J., Accomazzi, A.,& Grant, C. S. 1997, ``Historical Literature in the ADS,'' American Astronomical Society Meeting, 191, 3502

Eichhorn, et al. 1999
Eichhorn, G., Kurtz, M.J., Accomazzi, A., Grant, C.S., & Murray, S.S. 2000, ``The NASA Astrophysics Data System: The search engine and its user interface,'' A&AS, 143, 61

El-Ad & Piran 1997
El-Ad, H. & Piran, T. 1997, ``Voids in the Large-Scale Structure,'' ApJ, 491, 421
Garfield 1979
Garfield, E., 1979, ``Citation Indexing: Its Theory and Application in Science, Technology, and Humanities,'' New York: John Wiley
Genova et al. 1998
Genova, F., Bartlett, J. G., Bonnarel, F., Dubois, P., Egret, D., Fernique, P., Jasniewicz, G., Lesteven, S., Ochsenbein, F. & Wenger, M. 1998, ``The CDS Information Hub,'' ASP Conf. Ser. 145: Astronomical Data Analysis Software and Systems VII, 7, 470
Grant, Kurtz, & Eichhorn 1994
Grant, C. S., Kurtz, M. J.,& Eichhorn, G. 1994, ``The ADS Abstract Service: One Year Old,'' American Astronomical Society Meeting, 184, 2802
Grant et al. 2000
Grant, C. S., Eichhorn, G., Accomazzi, A., Kurtz, M. J., & Murray, S. S. 2000, ``The NASA Astrophysics Data System: Data holdings,'' A&AS, 143, 111
Helou & Madore 1988
Helou, G. & Madore, B. 1988, ``A new extragalactic database,'' Astronomy from Large Databases, ESO Conf. Proc. 28, eds F. Murtagh and A. Heck, p. 335
IEEE 1995
IEEE Computer Society 1995, ``1003-1995 IEEE guide to the POSIX Open System Environment,'' The Institute of Electrical and Electronics Engineers, Inc.
Jung 1971
Jung, J. 1971, ``Report on the Strasbourg Stellar Data Center,'' Bull. Info. Centre de Données Stellaires, 1, 2
Jung, Bischoff, & Ochsenbein 1973
Jung, J., Bischoff, M., & Ochsenbein, F. 1973, ``The catalog of stellar identifications,'' Bull. Info. Centre de Données Stellaires, 4, 27
Kristol & Montulli 1997
Kristol, D.M., Montulli, L., ``HTTP State Management Mechanism,'' RFC 2109, February, 1997
Kurtz et al. 2000
Kurtz, M. J., Eichhorn, G., Accomazzi, A., Grant, C. S., & Murray, S. S. 2000, ``The NASA Astrophysics Data System: Overview,'' A&AS, 143, 41
Loukides & Oram 1996
Loukides, M., & Oram, A. 1996, ``Programming With Gnu Software (Nutshell Handbook),'' O'Reilly & Associates, Inc.
Lynch 1997
Lynch, C. A. 1997, D-lib Magazine, Vol. 3, No. 4
Madore et al. 1992
Madore, B. F., Helou, G., Corwin, H. G., Jr., Schmitz, M., Wu, X. & Bennett, J. 1992, ``The NASA/IPAC Extragalactic Database,'' ASP Conf. Ser. 25: Astronomical Data Analysis Software and Systems I, 47
Marsden 1980
Marsden, B. G. 1980, Celestial Mechanics, 22, 63
Murray et al. 1992
Murray, S. S., Brugel, E. W., Eichhorn, G., Farris, A., Good, J. C., Kurtz, M. J., Nousek, J. A. & Stoner, J. L. 1992, Astronomy from Large Databases II, 387
Rood 1988
Rood, H. J. 1988, Voids, ARA&A, 26, 245
Salton & McGill 1983

Salton, G, & McGill, M. J. 1983, ``Introduction to modern information retrieval,'' New York, McGraw-Hill

Schatz & Hardin 1994
Schatz, B. R., Hardin, J. B. 1994, Science, 265, 895-901

Schmitz et al. 1995
Schmitz, M., Helou, G., Dubois, P., Lague, C., Madore, B., Corwin, H. G., Jr. & Lesteven, S. 1995, ``NED and SIMBAD Conventions for Bibliographic Reference Coding,'' ``Information & On-line Data in Astronomy,'' D. Egret & M.A. Albrecht, Eds., Kluwer Acad. Publ., 259

Tridgell 1999
Tridgell, A. 1999, ``Efficient Algorithms for Sorting and Synchronization,'' Ph. D. Thesis, The Australian National University

Vaudreuil 1992
Vaudreuil, G. 1992, ``MIME: Multi-Media, Multi-Lingual Extensions for RFC 822 Based Electronic Mail,'' ConneXions, 36-39

Wall, Christiansen & Schwartz 1996
Wall, L., Christiansen, T., & Schwartz, R. 1996, ``Programming PERL,'' O'Reilly & Associates, Inc., 2nd ed.

www.w3.org 1999
World Wide Web Consortium 1999, http://www.w3.org

Ylonen et al. 1999
Ylonen, T., Kivinen, M., Rinne, T., & Lehtinen, S. 1999, ``SSH Protocol Architecture,'' Internet Draft, Internet Engineering Task Force

Author Details

Guenther Eichhorn, Alberto Accomazzi, Carolyn S. Grant, Michael J. Kurtz and Stephen S. Murray
Harvard-Smithsonian Center for Astrophysics, Cambridge, MA 02138

For citation purposes:
G. Eichhorn et al., "The NASA Astrophysics Data System: Free Access to the Astronomical Literature On-Line and through Email", High Energy Physics Libraries Webzine, issue 5, Novembre 2001
URL: <http://library.cern.ch/HEPLW/5/papers/1>