

It's a year since the last release. This is a bit different from our nominal intention to have a release about every six months, but we have had a number of large developments to put in place. Also a big factor has been the degree of engagement with experiment and modelling programmes at the different laboratories – essential if the full effectiveness of shared analyses like CXSFIT and methodologies like superstages are to be realised. Allan, Martin and I seem to have been travelling a great deal. There has also been a re-location adjustment as I and part of the ADAS team are now based at UKAEA Culham/JET with only Allan and Chris still located at the Physics Department in Glasgow – although they are down at Culham on many occasions during the year.

Again one of the main drivers of our activities has been heavy species, with the superstage concept an essential part. Although a simple concept, the implementation has required a lot of adjustments so that the whole is largely transparent to the user. Some new items are unavoidable, such as the extension of the adf11 data classes to include effective charge (ZCD), effective squared charge (YCD) and effective ionisation potential (ECD). Trivial in the stage to stage framework, these are density and temperature dependent collisional-radiative quantities in the superstage framework and essential for fluid transport model consistency. From the ADAS user point of view, the key procedure is one which takes all the usual stage to stage adf11, adf15, adf40 files for an element and returns the same files but for a superstage. This procedure is ADAS416 and a brief description and some screen shots are shown later. For all this to be useful (and it is targetted at heavy elements), we do require a baseline set of adf04 files for all the heavy elements of interest. These are not all available yet, but do not be alarmed, they are on their way. Adam Foster has been doing a neat bit of work for their production, which regulates the structural complexity we adopt for each ion to available computer power and then gives details of missing power and missing transition arrays in configuration average approximation. He is at the mass generation phase now, so all this will go up very soon. You may at this point be completely bemused, but be not concerned. There is a large draft technical note (~70 pages) on the whole business of heavy species and superstages called 'Heavy species in fusion plasma modelling and spectroscopy' which we are just tidying up with Adam's final details, which I shall put in the ADAS documentation shortly. See item 1 below.

As I discussed a little at the ADAS workshop last year, it is interesting to consider the impact of heavier species in charge exchange spectroscopy. The expectation is that we shall progress from conventional individual spectrum line CXS at species up to argon or a bit beyond, with increasing difficulty, moving over to a many line weak CXS features from heavy species in various stages of ionisation. I do not think we can ignore this, especially if ultimately it remains an intention to dig slowing alpha CXS emission out of a Bremsstrahlung background. We have implemented a first stage attack on this problem with the new codes ADAS315 and ADAS316. ADAS316 takes us back to a very many bundle-n shell population model, of the sort used for beam deuterium beam stopping, but with charge exchange as the driving populating mechanism. It seemed an opportunity to revisit and improve the bundle-n codes so that they are more versatile and easier to use. This is an interesting situation. Bundle-n codes were originally developed for handling dielectronic recombination to very high n-shells. In fact as we go to very high charge states, the dielectronic capture tends to pull in somewhat to lower n-shells. It is charge exchange to more highly charged receivers from neutral hydrogen beams on the other hand that populates higher and higher n-shells, a situation compounded by the H(n=2) capture. ADAS316 needs an adf01 input dataset of the state selective capture cross-sections to each n-shell. This is a problem since we have such data only up to argon. At this point we have made a leap of faith based on z-scaling rules to obtain a universal z-scaled state selective data set, called ADF49, from which that of any individual ion may be recovered. ADAS315 does this extraction. There is a general pattern here, in which fiducial calculations on selected ions are used to underpin and optimised parameters of universal z-scaled approximants. I see this as a necessary route to go along with very heavy species which I hope to discuss more at the workshop. Another ADAS draft technical note 'Heavy species in charge exchange spectroscopy of fusion plasmas' is nearing completion which Adam is writing with me. See item 1 below.

The ADAS subroutine library documentation: Allan has been working on this large and hugely overdue update item – the revisiting of the ADAS subroutine libraries and the improvement of both access to them and the information for their use. Allan has put in a large effort on this, using the new methods he

has adopted for OPEN-ADAS, so that the library is self-documenting, searchable and so on. There are a few notes in item 2, but you should go and try it at our web site.

OPEN-ADAS was introduced in the last bulletin. Allan has been continuing work on it and it is planned for release outside the ADAS consortium at IAEA at the beginning of next year. Subroutines from the ADAS subroutine library which read data from the data formats in OPEN-ADAS will also be available from OPEN-ADAS.

CXSFIT now also has a complete manual which is out with readers and checkers. It will be put in the ADAS documentation. This past year really has seen CXSFIT worked vigorously in Europe. Allan has put much effort in to making small adjustments for the users and in ensuring the buffering interfaces to the specific machines are working OK. It has been a very successful effort in my view and gives encouragement that the ADAS Project can take on integrated analyses for shared use effectively. Our sights are now on NEW-CHEAP.

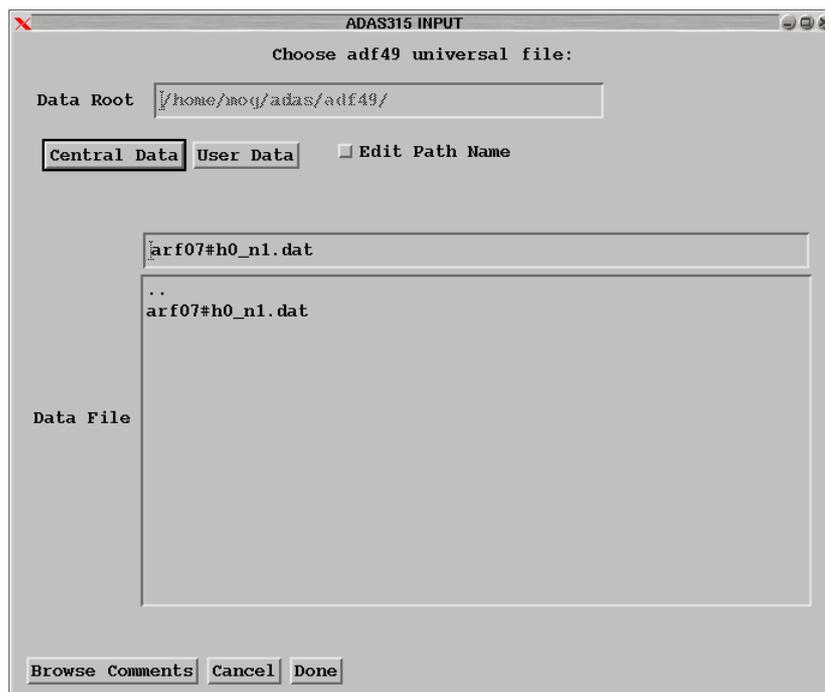
The list of code and data updates is in section 4. The list of code changes is very long, for which I apologise.

1. ADAS release 2.12

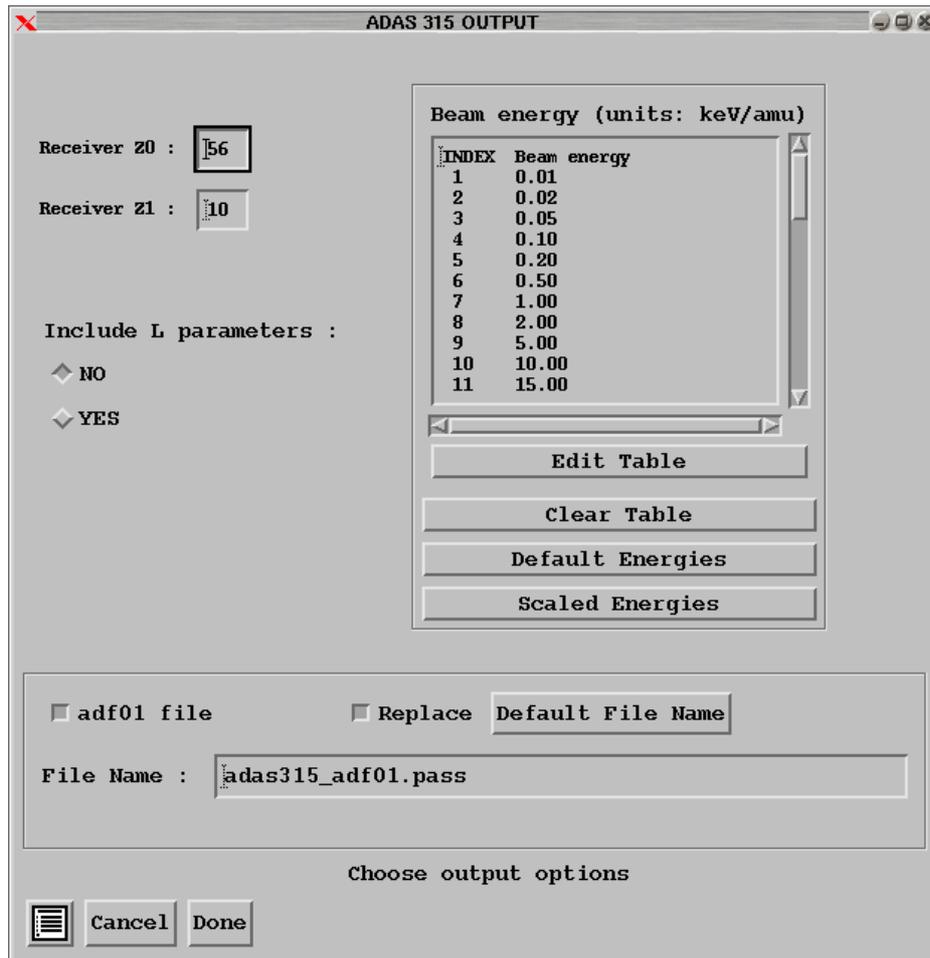
The new procedures are ADAS315, ADAS316 and ADAS416. Many subroutines have been added or modified to accommodate the new capabilities. The user should check the detailed release notes in item 4. A number of subroutines have modified parameter sets.

ADAS315: Preparation and extraction of universal adf49/adf01 CX cross-section data

This is a straightforward code which creates an ADF01 file for a specific ion of an element from a z-scalable universal dataset of format ADF49. There are only two datasets of ADF49 format corresponding to H(n=1) and H(n=2) donor respectively. The input screen is shown below followed by the output screen. A choice is possible of output energies. It is to be noted that in the level of approximation of the universal ADF49 data and the bundle-n application models, only the residual charge of the recombining ion is relevant. For the high n-shells of importance, the Rydberg electron is effectively in a hydrogenic state in the Coulomb field of the residual charge. The ADF49 datasets are subject to change as more fiducial data becomes available and the optimising of the z-scaling parameterisation is reworked. At this stage ADAS315 and ADAS316 are enabling a first look at the heavy ion CXS.

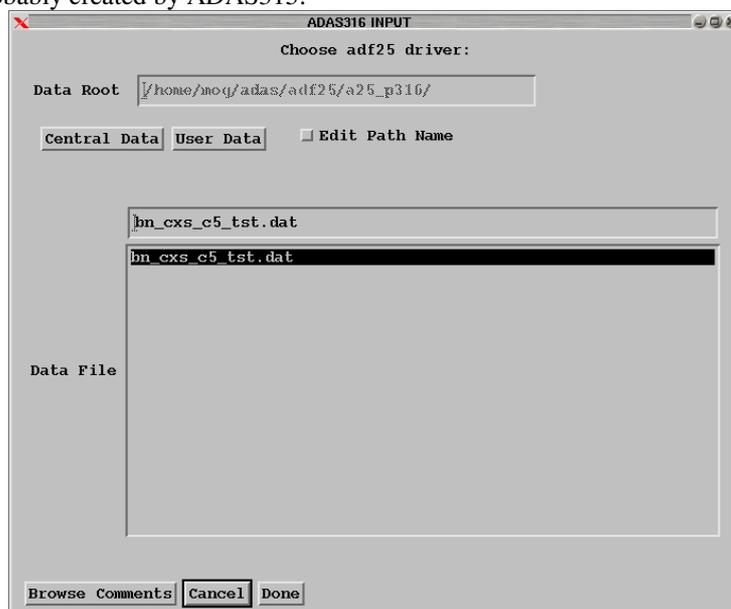


Running ADAS315 (an IDL only) code, available also at the command line, in a script allows output ADF01 files to be generated rapidly for a set of ions of an element.

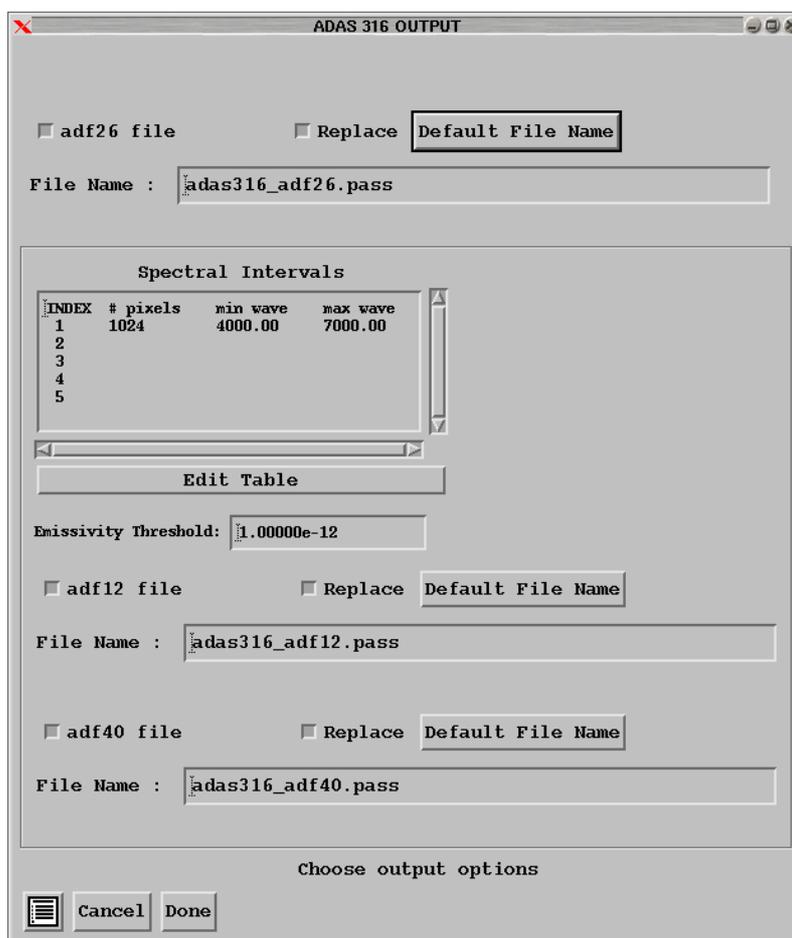


ADAS316: Charge exchange spectroscopy – process effective coefficients: bundle-n

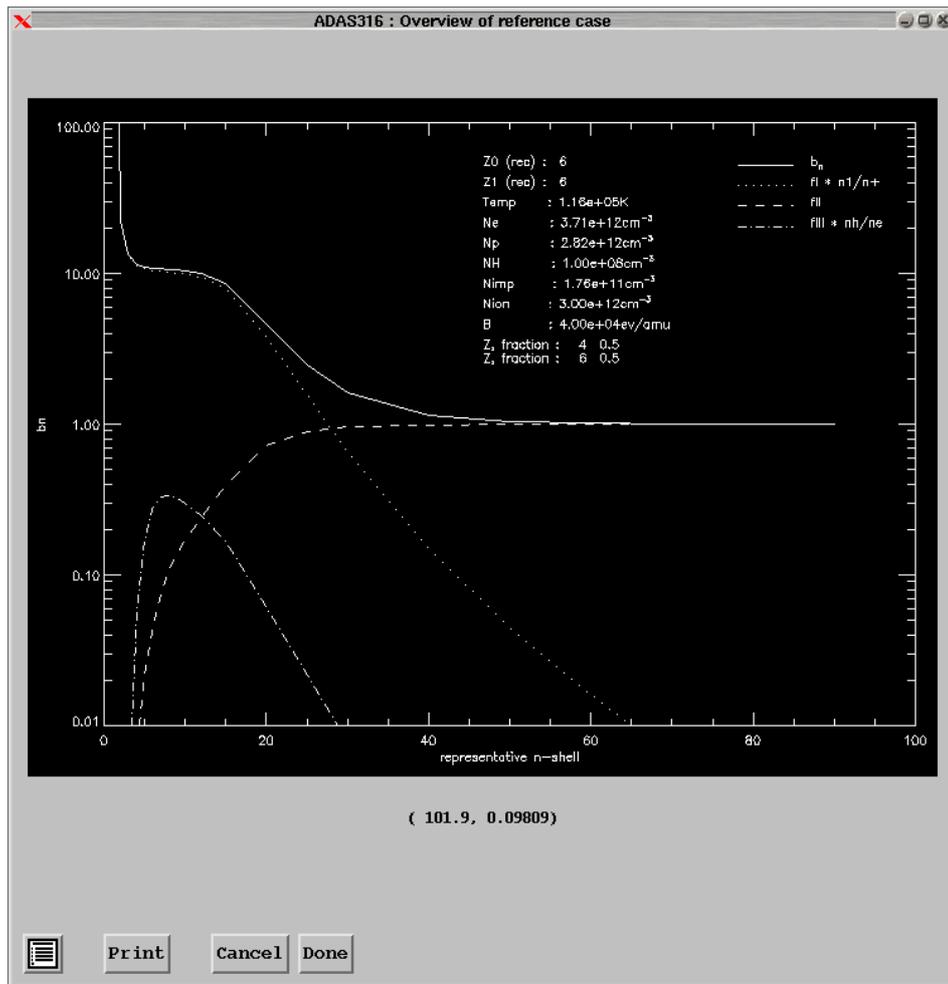
The code calculates charge exchange effective emission coefficients of format ADF12 from an input ADF01 file, probably created by ADAS315.



The code requires a driver data set and, for bundle-n in ADAS, these have historically been archived in ADF25. A new sub-directory /a25_p316 has been assigned and a complete redesign of the driver has been carried out. Also a new read routine exploits the new drivers fully. We would like to move to this type of driver in the future, where there is substantial guidance to the user in the fields and mnemonics used in the driver. The driver allows scans over parameters to be specified. In fact the driver is quite versatile and appropriate to all type of bundle-n calculation. We shall exploit it further for Bremsstrahlung + free-bound continuum and quasi-continuum high Rydberg line contributions in the future.



The output screen follows a usual pattern. Output ADF26 (the bundle-n population solution), ADF12 (charge exchange effective emission coefficients) and ADF40 (feature emissivity coefficients) may be produced. For heavy species CXS, because of the very large number of transitions between highly excited states, the ADF40 format becomes more useful than ADF12. A graph of the bundle-n population solution, at the reference plasma parameters (usually at the centroid of the scans) is generated and shown with the free-electron capture and charge exchange capture parts separated – as illustrated below.



ADAS416: Superstages – repartition adf11 and emissivity datasets

The interactive code, selected from the series 4 menu, follows the usual pattern. It starts with an input screen, which seeks a script for the superstage compression. These are allocated to the `./.../adas/scripts416` directory. It is expected that the user will have his/her own scripts, the central ADAS ones only being samples, such as shown for argon below. The partition specification starts at the root layer #01 in which each ionization is in its own partition. The layer #02 combines the stages in partitions at the user's wish. The choice of best partitioning is a subject of investigation in association with transport models, however the concept of 'natural partition' is helpful for the spectroscopist. The complete package for superstage compression includes a preview capability in IDL for the original and superstage ionization balance in natural partition, prior to the actual evaluation using ADAS416. The details are in the technical note.

```

argon
89
unresolved

parent pathways
-----
gcr : /home/summers/adas/adf11/**89/**89_ar.dat
pec : /home/summers/adas/adf15/pec06#18/ar_01/ic#ar**.dat
f-pec: /home/summers/adas/adf40/fpec06#18/ar_01/ic#ar**.dat

partition specification
-----
//#02/p00/ 00/p01/ 01/p02/ 02 03 04 05 06/p03/ 07/p04/ 08/
p05/ 09 10 11 12 13 14/p06/ 15/p07/ 16/p08/ 17/p09/ 18/
//#01/p00/ 00/p01/ 01/p02/ 02/p03/ 03/p04/ 04/p05/ 05/p06/ 06/p07/ 07/p08/ 08/
p09/ 09/p10/ 10/p11/ 11/p12/ 12/p13/ 13/p14/ 14/p15/ 15/p16/ 16/p17/ 17/
p18/ 18/
  
```

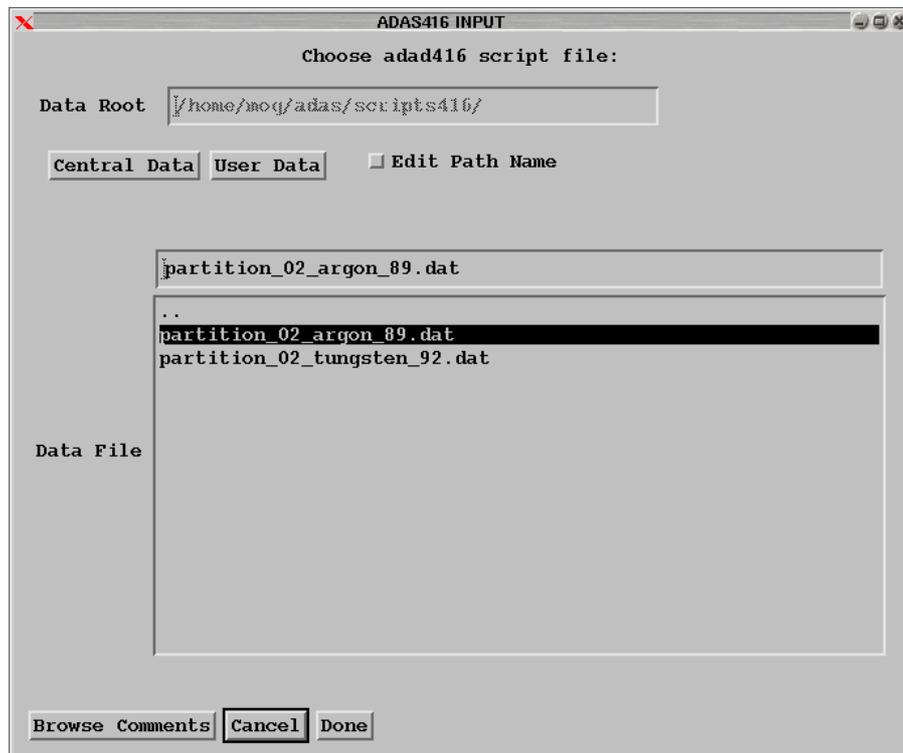
```

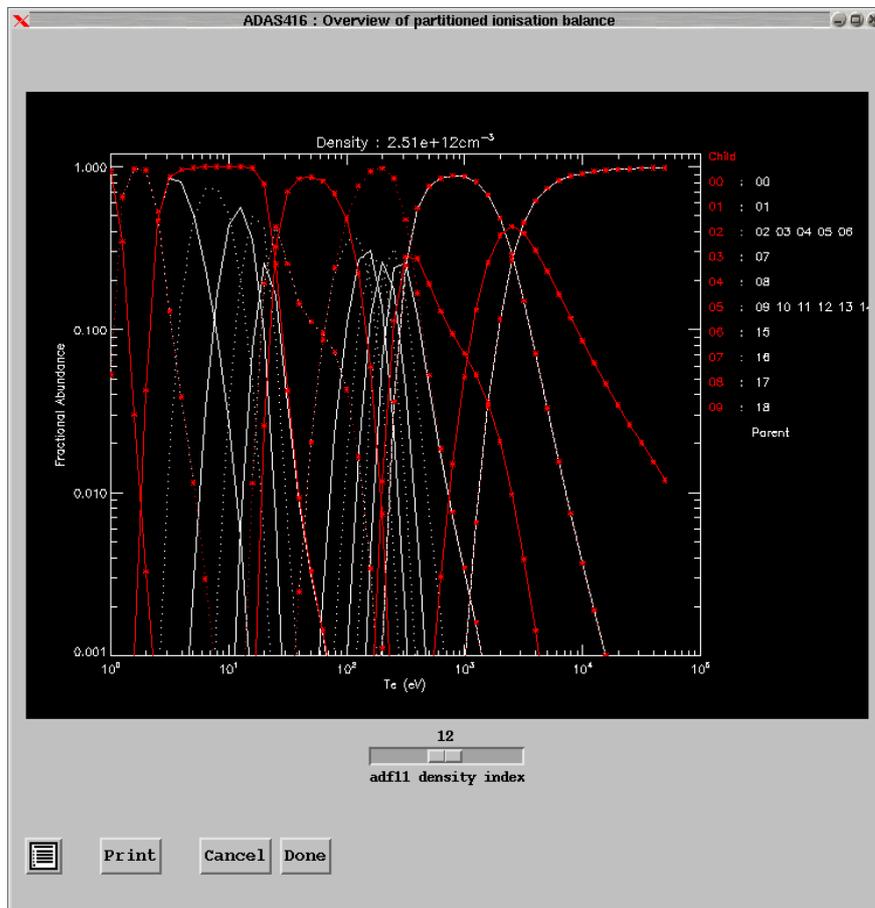
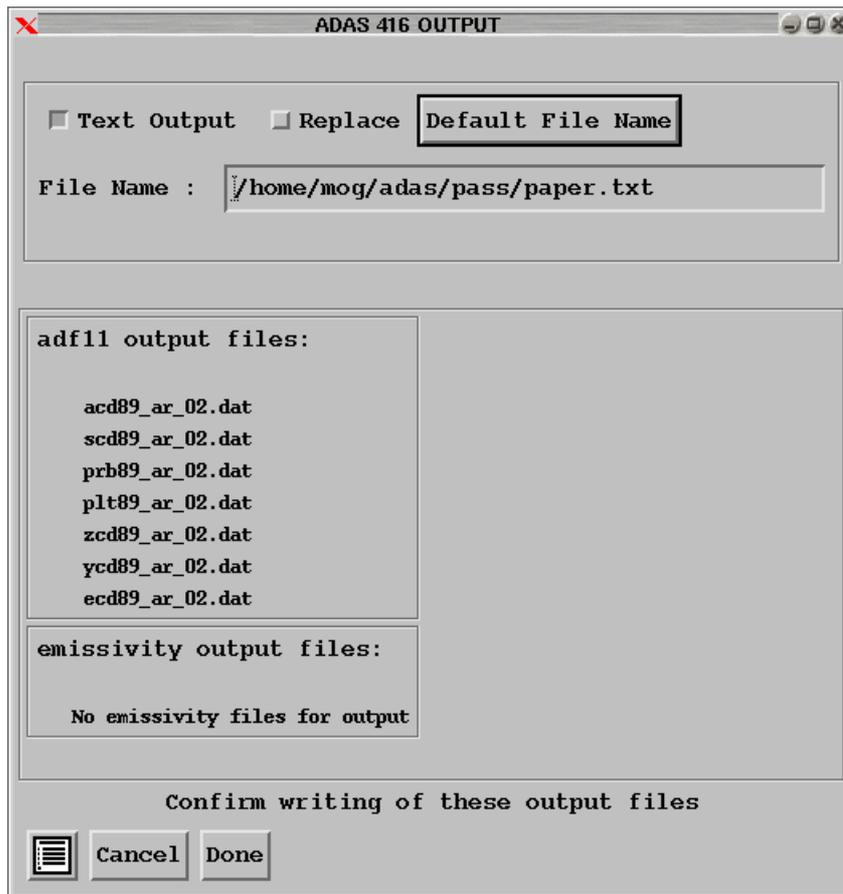
---
child pathways
-----
gcr : /home/summers/adas/adf11/**89/**89_ar_02.dat
pec : /home/summers/adas/adf15/pec06#18/ar_02/ic#ar**.dat
f-pec: /home/summers/adas/adf40/fpec06#18/ar_02/ic#ar**.dat

C-----
C sample scrip416 file
C
C Notes:
C (1) '***' in gcr pathway denotes 'acd', 'scd' etc
C (2) '***' in pec, gtn, f-pec and f-gtn pathways denote 'llr', 'pjr'
C or 'pju'.
C (3) '**' in pec, gtn, f-pec and f-gtn pathways denote partition
C member index. This is the ion charge for the #01 root
C partition. The indexing begins at 00
C (4) Leave pathways which are not needed as blanks
C (5) Choices at line 4 are 'resolved' or 'unresolved'
C (6) Partitions are given in decreasing level order, that is
C child before parent. This is the same rule as is used in
C adf11 files with partition blocks and is required by the
C partition block reader xxrptn.for.
C
C
C Author: Hugh Summers
C Date : 18 august 2005
C-----

```

The input screen is shown below and, following *Done* leads directly to the output screen where the options of writing the new partition are given. This screen reacts to the available ADF11 files for the parent partition. Note the new classes *zcd*, *yed* and *ecd*. The output ADF11 files uses an extended specification which includes partition and connection vector information at the head of the file. New ADF11 reading routines are in place in the ADAS libraries to which deal with the extended formatting as well as reading older versions of ADF11.

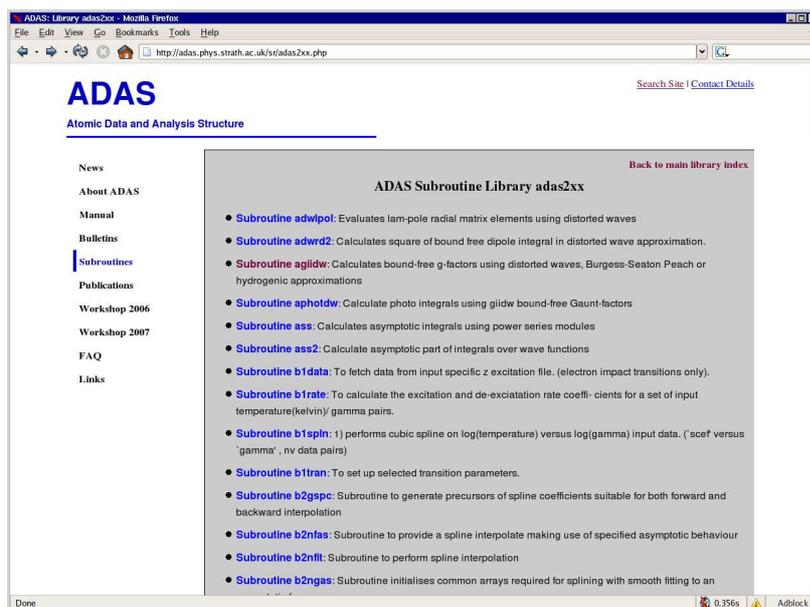




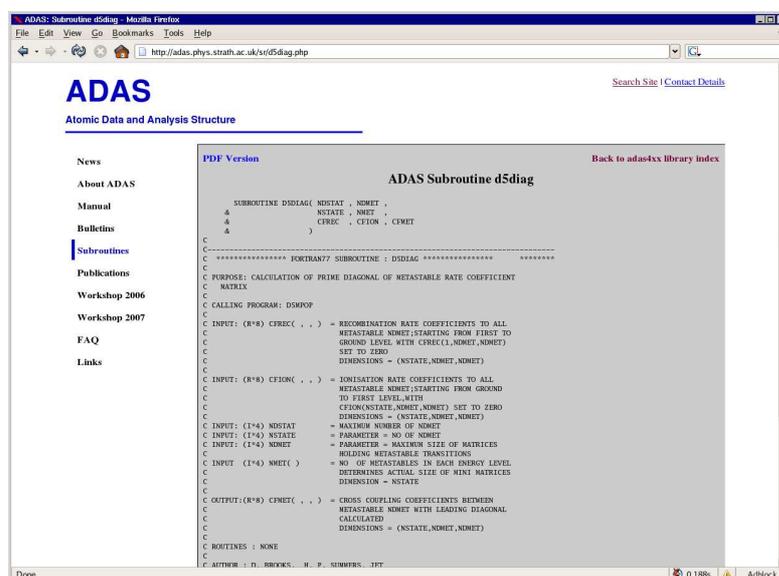
2. ADAS Subroutine Library Documentation

Allan has done a complete revision of appendix B of the manual - this is the documentation for the various Fortran subroutines. We now have every subroutine in our library documented along with the various declarations which are required to call the subroutine. Appendix B now stretches to 1,565 pages and is available as a PDF document in the doc/manual directory of ADAS.

In addition, all of the subroutines are also on the web. From the default ADAS web page, select the 'Subroutines' option at the left hand side and you can browse through each subroutine and read its description – it is also possible to download single PDF documents for individual subroutines from this site. Of course, all of the documentation can be searched using the search function on the site - available from the link at the top right of the screen.



At the end of appendix B, examples are given of calling subroutines from both C and Fortran including a simple Bremstrahlung call as well as a more elaborated ionisation balance calculation using the core routines from ADAS405. Preparation of this document was a major undertaking so please let us know of any errors you happen to find. We are going to follow this up with an IDL version for the next release.



3. CXSFIT

CXSFIT is now being used routinely on JET and AUG for all core CX systems. Very encouraging work has also been done on the edge systems of both machines. Initial testing at TEXTOR went well and full integration with the TWU and TPD systems was achieved. and we expect it to become the default code at TEXTOR.

4. Code and data updates in v2.12.

Corrections and updates to code (ADAS v2.11 to ADAS v2.12)

C.1 Added

nsort.for
xxdeci.for
xxfrmt_t.for
xxfrmt_trm.for
xxmkrc.for
xxmkrp.for
xxwcmt_15.for
xxwcmt_40.for
to offline_adas/adaslib

C.2 Changed offline version of ADAS810:

Altered output on header lines for superstage compatibility.
Altered strategy for power ranking of emissivities
Altered comment lines for superstage compatibility
and field key reading of comments.
Modified for new xxdata_04 routine

C.3 Update xxdata_04 and xxprs1 in offline_adas only to handle (Jp)j coupling At the same time, brought xxdata_04 in offline ADAS up to date with changes in interactive ADAS.

C.4 Modifications of ifgpp stage of adas8#1 to:

Allow more flexible spacing in adf04 files (a04filter.for)
Carry Sval of level in unused part of labels (levelread.for, termread.for)
Organisational changes to output (ifgpp.for)

C.5 Addition of run_adas810 to adas8#1 as a wrapper to the 810 stage of adas8#1

C.6 Update of h8nlev, extension up to the 8k (i.e. n=8 l=7) shell and capability for term_count as an output.

C.7 Update of occ2cow to deal more generally with the subtleties of input to the Cowan code.

C.8 Updated xxdata_40 to read new partitioned datasets, complete re-write of IDL wrappers was also done. read_adf40.pro was upgraded to use this version.

C.9 A fortran xxdata_00.for, to access the adf00 ground configuration and ionisation potentials of arbitrary ions, is now available.

C.10 ADAS408 upgraded to produce new adf11 classes:

ZCD - effective ion charge
YCD - effective squared ion charge
ECD - effective ionisation potential.

The interactive and IDL version (run_adas408) can generate these new adf11 datasets.

- C.11 Added `adas_string_pad`: a utility code to pad strings or string arrays out to a given length. Used by IDL wrapper to `xxdata_40`.
- C.12 Updated `xxname.for` to allow for USERIDs up to 20 characters, previous limit was 8 (the nominal limit for unix) but modern unix distributions allow longer USERIDs.
- C.13 Numerical errors in `adas509` which showed up on Sun systems have been trapped. A correct `adf14` file, without NaNs, is now produced.
- C.14 Some fixes to `adas404` which only affect neon `adf11` datasets.
- F-like and Ne-like stages have two metastables, not one.
 - K/eV conversion factor now uses same value as `xxtconv` routine.
 - Internal `erg/J` conversions check if value is zero and sets an effective zero of $1\text{D}-74$ rather than $1\text{D}-7$.
- C.15 Corrected the standard number of metastables for Ne+0 to 2. This affects subroutines `xxdata_40.for`, `xxmkrc.for` and `xxmkrc.for`.
- C.16 Introduced a substantially revised version of `xxdata_00.for` which allows reading of standard and metastable LS resolved forms of `adf00` datasets.
- C.17 Modified ADAS408 for compatibility with revised `xxdata_00.for`. Altered codes are `adas408.for`, `d8out1.for`, `d8wzcd.for` and the compilation script
- C.18 Added `xxdata_11.for` and corresponding IDL procedure `xxdata_11.pro`.
- C.19 Fixed bug in `offline_adas ADAS8#3` script where `datestamp` was off by one month.
- C.20 Added `/help` keyword to `run_adas208`
- C.21 Modifications to `cxsqef.pro` (affecting `read_adf12.pro`) to work with g77 Linux systems.
- C.22 Changes to `offline adas810`:
- There was an internal conflict in opening files if both a filter (`adf35`) and projection data (`adf18/a17_p208`) were chosen. `Offline adas810` will not happily accept both options.
 - The comments for non selected output files are no longer written so random `fort.?` files are not produced.
 - An internal dimension problem has been fixed in the population routine which could give incorrect handling of ionisation from excited states. Note that this problem is not in the interactive `adas810`.
 - Bring interpolation of filter data into line with central version (`d8tran.for`).
- C.23 Remove some superfluous routines from `adas414` which may interfere with user's code as the names were too generic. In detail `onoff.pro`, `filter.pro`, `plot_henke.pro` and `phenke.pro` will no longer appear in the user's IDL path.
- C.24 In the utility routine, `period_system`, change the indexing of array elements from the obsolete round parenthesis to the square bracket.
- C.25 Allow lowercase `d` and `e` as acceptable parts of a number in the `r8fctn.for` parsing routine.
- C.26 Add `fulldata` as an optional output to `read_adf11.pro` which uses the new `xxdata_11` reading routine.
- C.27 In `read_adf01` the maximum number of n-shells permitted is increased from 20 to 100 in order to accommodate handling of high Z species. In line with naming the `adf` reading routines with `xxdata_<no>` this uses `xxdata_01`.

The output of the fulldata structure has been slightly modified. The n, nl and nlm cross sections are returned up to the maximum n in the dataset. However the lower n is always 1, not nmin. The appropriate cross section index can be found using the indexing functions i4idfl.pro and i4idfm.pro. Use the optional /idl_index keyword to keep the IDL 'first index is 0' convention.

C.28 Removed matin1.for from fortran/adas1xx/adas101 and from fortran/adas1xx/adas108. These were duplicates of routines in fortran/adaslib/maths and should never have been present.

C.29 Moved adaslib/maths/{form,form2,form4}.for to adaslib/atomic.

C.30 Moved adaslib/atomic/solve.for to adas2xx/adas203/b3solv.for

C.31 Moved adaslib/atomic/r8ovlp.for to adas3xx/adaslib/cxovlp.for

C.32 Moved adaslib/atomic/r8f21.for to adaslib/maths/r8f21.for

C.33 Small (mainly formatting) changes to the comments sections of:

- adaslib/atomic/bf.for
- adaslib/atomic/i4idfl.for
- adaslib/atomic/i4idfm.for
- adaslib/atomic/i4idli.for
- adaslib/atomic/i4jgam.for
- adaslib/atomic/r8adif.for
- adaslib/atomic/r8dcon.for
- adaslib/atomic/r8ecip.for
- adaslib/atomic/r8econ.for
- adaslib/atomic/r8fdip0.for
- adaslib/atomic/r8gam.for
- adaslib/atomic/r8gav.for
- adaslib/atomic/r8gbf.for
- adaslib/atomic/r8giiav.for
- adaslib/atomic/r8giii.for
- adaslib/atomic/r8giiiav.for
- adaslib/atomic/xxceia.for
- adaslib/atomic/xxdtes.for
- adaslib/maths/matinv.for
- adaslib/maths/r8fun1.for
- adaslib/maths/r8fun2.for
- adaslib/maths/xxder1.for
- adaslib/maths/xxeign.for
- adaslib/maths/xxhunt.for
- adaslib/maths/xxmnmx.for
- adaslib/maths/xxnbaf.for
- adaslib/maths/xxsple.for
- adaslib/maths/xxsplf.for
- adaslib/maths/xxspln.for
- adaslib/maths/xxycf2.for
- adaslib/system/xxin17.for
- adaslib/system/xxin80.for
- adaslib/system/xxinst.for
- adaslib/utility/i4fctn.for
- adaslib/utility/i4indf.for
- adaslib/utility/i4indfi4.for
- adaslib/utility/i4indfvs.for
- adaslib/utility/lenstr.for
- adaslib/utility/r8fctn.for
- adaslib/utility/xxfchr.for
- adaslib/utility/xxgama.for
- adaslib/utility/xxlim4.for

adaslib/utility/xxlim8.for
adaslib/utility/xxlm28.for
adaslib/utility/xxpars.for
to allow the automation of subroutine documentation.

C.34 Added documentation to:

adaslib/atomic/r8fdip.for
adaslib/atomic/r8fdip1.for
adaslib/atomic/r8fdip2.for
adaslib/atomic/r8fmon1.for
adaslib/atomic/r8p.for
adaslib/atomic/r8prov.for
adaslib/atomic/r8qp.for
adaslib/atomic/r8zeta.for
adaslib/atomic/z16cd.for
adaslib/math/ee2.for
adaslib/math/ee3.for
adaslib/math/eei.for
adaslib/math/r8f21.for
adaslib/math/r8feei.for
adaslib/math/r8xip.for
adaslib/math/r8yip.for

C.35 Moved adaslib/atomic/z16cd.for to adas3xx/adaslib/z16cd.for.

C.36 Moved adaslib/math/r8xip.for and adaslib/math/r8yip.for to adaslib/atomic.

C.37 Moved adaslib/atomic/form.for to adas2xx/adas203/b3form.for and adaslib/atomic/form2.for to adas2xx/adas203/b3form2.for

C.38 Added /upper and /lower keywords to xxesym to return element symbol in either full lowercase of full uppercase respectively.

C.39 A dimension for splining in ceevth.pro is now set dynamically to the maximum of the input or output array. Previously the default was 40 which may not be sufficient.

C.40 Updated comments on:

adas1xx/adas108/a8afit.for
adas1xx/adas106/lspij3.for
adas1xx/adas106/lsort.for
adas1xx/adas106/lgspc.for
adas1xx/adas106/lgasym.for
adas1xx/adas106/lfitsp.for
adas1xx/adas106/lfasym.for
adas1xx/adas105/spfman11.for
adas1xx/adas103/spfman8hx.for
adas1xx/adas103/nspij3.for
adas1xx/adas103/ngspc.for
adas1xx/adas103/ngasym.for
adas1xx/adas103/nfitsp.for
adas1xx/adas102/spfman5e.for
adas1xx/adas103/burgfs.for
adas1xx/adas103/dnsort.for
adas1xx/adas103/nfasym.for
adas1xx/adas106/spfman12.for
to allow for automatic documentation generation.

C.41 Updated comments on:

adas2xx/adas204/b4proj.for

adas2xx/adas204/ngffmh.for
adas2xx/adas211/adwlpol.for
adas2xx/adas211/adwr2.for
adas2xx/adas211/agiidw.for
adas2xx/adas211/aphotdw.for
adas2xx/adas211/bdcf7.for
adas2xx/adas211/dnaq.for
adas2xx/adas211/rd2fs.for
adas2xx/adas215/bfdata.for
adas2xx/adas216/bgrerr.for
adas2xx/adas208/b8data.for
adas2xx/adas209/b9data.for
adas2xx/adas210/badata.for

to allow for automatic documentation generation.

C.42 Updated comments on:

adas8xx/adas804/diag.for
adas8xx/adas804/dipol.for
adas8xx/adas804/dipsum.for
adas8xx/adas804/rbesf.for

to allow for automatic documentation generation.

C.43 Updated comments on:

adas7xx/adaslib/gximpr.for

to allow for automatic documentation generation.

C.44 Updated comments on:

adas4xx/adas404/d4open.for
adas4xx/adas404/lh404ru.for
adas4xx/adas407/d7auts.for
adas4xx/adas407/d7cors.for
adas4xx/adas407/fmin.for
adas4xx/adas407/gpcalc.for
adas4xx/adas407/gpcalcx.for
adas4xx/adas407/gpcall.for
adas4xx/adas407/lmdif1_all.for
adas4xx/adas407/nvgoel.for
adas4xx/adas407/rbchid.for
adas4xx/adas412/adas408/d8gpca.for
adas4xx/adas412/adas408/d8vgol.for
adas4xx/adas412/adas408/ioadas1.for
adas4xx/adas412/adas408/ioadas2.for
adas4xx/adas412/c1u2lc.for

to allow for automatic documentation generation.

C.45 Corrected typo in comments of:

adas3xx/adas306/c6qeik.for
adas3xx/adaslib/cxqeik.for

C.46 Updated comments on:

adas3xx/adas302/sigia.for
adas3xx/adas304/c4mnmx.for
adas3xx/adas305/bornp1.for
adas3xx/adas305/bornp2.for
adas3xx/adas305/drv.for
adas3xx/adas305/fsplin.for
adas3xx/adas305/hydemi.for
adas3xx/adas305/lumsis.for
adas3xx/adas305/sfi2.for

adas3xx/adas305/sigma.for
adas3xx/adas305/sigmel.for
adas3xx/adas305/sigmin.for
adas3xx/adas306/c6qeik.for
adas3xx/adas306/c6qxch.for
adas3xx/adas306/c6tbfm.for
adas3xx/adas306/c6tbqm.for
adas3xx/adas310/abi.for
adas3xx/adas310/abinew.for
adas3xx/adas310/bnqctb.for
adas3xx/adas310/casszd.for
adas3xx/adas310/catmpf.for
adas3xx/adas310/cldlbn2.for
adas3xx/adas310/lowpop.for
adas3xx/adas310/nsuph1.for
adas3xx/adas310/pypr.for
adas3xx/adas310/pyvr.for
adas3xx/adas310/qlpr.for
adas3xx/adas310/qpr78.for
adas3xx/adas310/qvain.for
adas3xx/adas310/rqinew.for
adas3xx/adas310/rqlnew.for
adas3xx/adas310/rqvnew.for
adas3xx/adas311/ghnle.for
adas3xx/adas311/ghnlv.for
adas3xx/adas311/lftime.for
adas3xx/adas311/matintl.for
adas3xx/adas311/nhydes.for
adas3xx/adas311/nlthes.for
adas3xx/adas311/nwripv.for
adas3xx/adas311/omgrc2.for
adas3xx/adas311/rd2bs.for
adas3xx/adas311/rmdegv.for
adas3xx/adas311/rpengv.for
adas3xx/adas311/start7.for
adas3xx/adas311/supphe1.for
adas3xx/adas314/ceecon.for
adas3xx/adas314/ceevth.for
adas3xx/adas314/cefill.for
adas3xx/adas314/ceparm.for
adas3xx/adas314/cesgcx.for
adas3xx/adas314/cewr11.for
adas3xx/adas314/cewr12.for
adas3xx/adaslib/cxcrdg.for
adas3xx/adaslib/cxcrip.for
adas3xx/adaslib/cxmrdg.for
adas3xx/adaslib/cxqeik.for
adas3xx/adaslib/cqxch.for
adas3xx/adaslib/cxsqef.for
to allow for automatic documentation generation.

C.47 Removed non-standard control character from comments of:
adas2xx/adas211/bbprs3.for

C.48 Updated comments on:
adas5xx/adas509/sigcx.for
to allow for automatic documentation generation.

C.49 Major revision of list of subroutines included in static library

C.50 The move to high Z for charge exchange has prompted the removal of the m-subshell data possibility from the adf01 datasets. The series 3 codes now use the altered xxdata_01.for routine which now returns the n and nl cross sections only. In reality there was no dataset in ADAS with m-subshell data.

The IDL access routine, read_adf01.pro and xxdata_01.pro have also been brought into line.

The affected routines are:

```
fortran/adas3xx/adas301/adas301.for
fortran/adas3xx/adas301/c1bsig.for
fortran/adas3xx/adas306/adas306.for
fortran/adas3xx/adas307/adas307.for
fortran/adas3xx/adas308/adas308.for
fortran/adas3xx/adas309/adas309.for
fortran/adas3xx/adas310/bnqctb.for
fortran/adas3xx/adas314/adas314.for
fortran/adas3xx/adas314/cefill.for
fortran/adas3xx/adas314/cether.for
fortran/adas3xx/adas314/cewr11.for
fortran/adas3xx/adas314/cewr12.for
fortran/adas3xx/adas314/comp314
fortran/adas3xx/adaslib/cxextr.for
fortran/adas3xx/adaslib/cfrac.for
fortran/adas3xx/adaslib/cxqcx.for
idl/adas3xx/adas301/adas301.pro
idl/adas3xx/adas301/cw_adas301_proc.pro
idl/read_adf/read_adf01.pro
idl/read_adf/xxdata_01.pro
wrapper/adaslib/xxdata/xxdata_01_calc.for
wrapper/adaslib/xxdata/xxdata_01_if.c
wrapper/read_adf/read_adf01/read_adf01.c
wrapper/read_adf/read_adf01/readadf01.for
```

C.51 Also as part of this move the adf12 dataset format has been modified. The significant changes are that the 1-D scanning arrays are no longer padded with zeros and the information in the block identifier line has been improved. Now the wavelength of the transition is returned (if present).

Consequently the access routine, xxdata_12.for, xxdata_12.pro and read_adf12.pro have been altered. These will now read both new and old style output files. Note that codes calling the xxdata_12 variants will need to be altered to reflect the extra (element, ion charge and wavelength) information returned. The fulldata structure has been updated in read_adf12.pro and when called in interpolation mode the wavelength is another optional output.

C.52 The change to xxdata_12 has a knock-on effect on other ADAS codes. cxsqef.for and cxsqef.pro required alteration.

cxsqef.pro now issues a warning if called with the old parameter set asking the user to change the calling routine. However the routine will still work and return the correct qeff values.

Unfortunately the fortran code is not amenable to such a fix and all fortran codes which use it will have to be recompiled. It is never ideal to change the API of a routine but the increased flexibility and wavelength information necessitates the change.

C.53 An IDL version of the free-free Maxwell average Gaunt factor, <giii>, is available as r8giiiav.pro (as a function).

C.54 Use of CHAR(0) as C style string terminator rather than '\0' in xxname.for to provide forward compatibility with fortran 90 (specifically gfortran).

C.55 An IDL routine to write adf15 PEC files (write_adf15.pro) has been added. As with the other write_adfXX routines this takes the fulldata structure as its input.

C.56 A new interactive code, adas316, is introduced. This prepares effective emission coefficient data in the bundle-n approximation. The data is archived as adf12 datasets and optionally as an adf40 feature emissivity.

Note that the code uses a revised v2b1mod for heavy species charge exchange. It encompasses the old looped running of v2bnmod (replaced by now by cgbnhs) and the primary output reprocessing to adf26 tables which has now been replaced by direct output to adf12. It is designed for use of universal adf49 sourced CX data (from adas315) and application to partially ionised heavy specie receivers.

As part of implementing adas316 the underlying routine (cldlbn2.for) acquired a new argument to control output of passing data because adas310 manages file output differently. Both codes can now use it.

C.57 adas316 works on adf25 driver files. These are different to the versions used in adas204 (as generated by adas407) and will be archived in the .../adas/adf25/a25_p316/ directory. The format is much more transparent than the existing one, which will be brought into line.

A new fortran routine, xxdata_25.for, and an accompanying xxdata_25.pro are included. For completeness a read_adf25.pro and a write_adf25.pro which operate on fulldata structures are provided.

C.58 There was an indexing error in the hapixv.for routine which distributed emission data onto a pixellated grid. Both the online and offline versions of adas810 have been updated.

This error also affected c5pixv and the IDL c5dplr routine has been updated.

A general purpose routine, xpixv.for, has been added to the adaslib library.

C.59 A new interactive code, adas315, is introduced. This takes universal scaled charge exchange cross section data (in a new adf49 format) and produces individual adf01 files for arbitrary element and charge state. adf49 data for hydrogen n=1 and n=2 donor information is included in this release.

The core of the code, run_adas315.pro, can be run independently if called from the user's IDL codes. This code has been designed in association with adas316 to allow full scripting for large scale data production.

As part of this code additional routines, which may be of more general use, were required. read_adf49.pro, xxdata_49.pro and write_adf01.pro are now available. All use the conventional fulldata structure method to pass data.

Note that adas315 does not contain any fortran code.

C.60 The utility xxsp1e.pro routine passed back the interpolated rather than the extrapolated points in the extrap vector. Add a complementary interp output variable to cover both cases.

C.61 Increased dimensions of ADAS205 - number of transitions is now 250,000.

C.62 Removed routine:

adas2xx/adas211/rrcqd1.for

since it was just an older version of:

adas2xx/adas211/bbrint.for

C.63 Removed comment from bbrint.for which implied it couldn't do Druyvesteyn and numerical distributions.

C.64 Small modifications in comments of routines:

adas2xx/adas204/b4flnm.for
adas2xx/adas207/b7lrat.for
adas2xx/adas208/b8corp.for
adas2xx/adas208/b8corr.for
adas2xx/adas208/b8loss.for
adas2xx/adas208/b8totl.for
adas2xx/adas208/r8necip.for
adas2xx/adas211/dnaq.for
adas2xx/adas211/gext.for
adas2xx/adas211/pchg.for
adas2xx/adas211/pchx.for
adas2xx/adas211/phase.for
adas2xx/adas212/isort.for

to allow for automatic documentation preparation.

C.65 Removed junk from columns > 72 in adas2xx/adas212/isort.for.

C.66 Modifications in comments of routines:

adas5xx/adas501/e1chkb.for *
adas5xx/adas501/e1file.for *
adas5xx/adas502/sszd.for *
adas5xx/adas503/e3chkb.for *
adas5xx/adas504/spzd.for *
adas5xx/adas507/e7chkz.for *
adas5xx/adas509/e9chkb.for *
adas5xx/adas509/e9chkz.for
adas5xx/adas509/e9data.for

to allow for automatic documentation preparation, also added more standard revision histories for the ones marked with stars, best guesses of revisions were obtained from historical SCCS archives.

C.67 Add an IDL version of r8yip (evaluates the impact parameter cross-section second Bessel integral Y) for those who want to explore ECIP cross sections and rates in IDL.

C.68 xxdata_15.for is extended to deal with superstages. The IDL routines, xxdata_15.pro and read_adf15.pro, have been updated to use the new version. More fields are returned in the fulldata structure and the method of determining the wavelength of a transition is more robust.

C.69 Interactive adas810 now writes the extended adf15 and adf40 datatypes (for superstages and partitioning) following its use in the offline version.

C.70 An IDL command, run_adas311.pro, is provided to allow non-interactive use of adas311.

C.71 Add an IDL version of the adas xxminv library routine which performs matrix inversion with accompanying solution of linear equations if requested.

C.72 Increased version number to 2.12.

Corrections and updates to data (ADAS v2.11 to ADAS v2.12)

D.1 Aluminum has been added to the system. Baseline adf04 data at LS and IC resolutions is in adf04/copmm#13. New 89 style adf11 data along with the adf03 driver file is also added.

D.2 Add charge exchange recombination rate data for H and D with a deuterium donor. This complements the T+ + D0 rate already present in adf11/ccd96 directory.

The associated prc data should be zero as in a charge exchange interaction between indistinguishable reactants no excess power is expected. The values present in the H and T datasets were an artifact of the previous post-processing method. The data in adf11/prc96 have now been set to zero.

- D.3 Update of all ionisation potentials in adf00/w.dat from NIST Paper of A E Kramida, J Reader, At. Dat. and Nucl. Dat. 92 (2006) 457-479
- D.4 Add metastable resolved adf00 datasets in LS coupling for elements H, He, Li, Be, B, C, N, F, Ne and Ar. The datasets include excitation energies in addition to ionisation energies as required in the preparation of resolved adf11/ecd data. The datasets take the form '<elem symb>_ls.dat'.
- D.5 Add the donor mass (deuterium) to the block identifier line in the 96 adf11 charge exchange data collection. Specifically the following datasets have been altered.

adas/adf11/ccd96r/ccd96r_c.dat
adas/adf11/ccd96/ccd96_c.dat

adas/adf11/prc96r/prc96r_c.dat
adas/adf11/prc96/prc96_c.dat
adas/adf11/prc96/prc96_d.dat
adas/adf11/prc96/prc96_h.dat
adas/adf11/prc96/prc96_he.dat
adas/adf11/prc96/prc96_t.dat

Although not a change in data this value was omitted from the original calculation so the datasets have been changed rather than superseded.

- D.6 A few typos crept into the ionisation potentials of Cs, Dy, Lu and Ba in the adf00 archive. Spotted by the IPP modellers!
- D.7 The charge exchange cross section data for Z elements (adf01 datasets) can be generated with the new adas315 code. Two universal scaled datasets for hydrogen n=1 and n=2 donors are archived in a new datatype, adf49. Adam Foster has compiled both and they are named:
- adas/adf49/arf07#h0_n1.dat
adas/adf49/arf07#h0_n2.dat
- D.8 PEC data for two Fe ionisation stages (Fe23+ and Fe14+) has been added to the adf15/transport/ directory. This collection is to support transport analysis. The input data is from collision calculations by Allan Whiteford and Don Griffin but is outside the GCR framework for the time being.

HPS
7 Oct. 2007