

INDC International Nuclear Data Committee

Summary Report from the Consultants' Meeting on GNDS/FUDGE/TAGNDS

IAEA Headquarters, Vienna, Austria

22-25 May 2023

Prepared by
Caleb Mattoon (USA) and
Jean-Christophe Sublet (IAEA Nuclear Data Section)

July 2023

Selected INDC documents may be downloaded in electronic form from
<http://www-nds.iaea.org/publications>
or sent as an e-mail attachment.

Requests for hardcopy or e-mail transmittal should be directed to
NDS.Contact-Point@iaea.org

or to:

Nuclear Data Section
International Atomic Energy Agency
Vienna International Centre
PO Box 100
1400 Vienna
Austria

Printed by the IAEA in Austria

July 2023

Summary Report from the Consultants' Meeting on GNDS/FUDGE/TAGNDS

IAEA Headquarters, Vienna, Austria

22-25 May 2023

Webpage: <https://conferences.iaea.org/event/352/>

Prepared by
Caleb Mattoon (USA) and
Jean-Christophe Sublet (IAEA Nuclear Data Section)

July 2023

Contents

Objectives & white paper	7
Applications.....	7
YANDF: An attempt to unify TALYS, EXFOR and ENDF data formats	8
Developing a Web Service for EXFOR Using RESTful API and JSON Schema	8
TAGNDS and Autotalys update	10
LLNL’s FUDGE and GIDI+ Code Packages	11
The FUDGE translation of the TENDL-2021 library from ENDF-6 to GNDS.....	12
Handling of GNDS in FRENDY and JAEA recent activity	13
NJOY2016 modernisation and GNDS support in scion	13
Recommendations	15
References	17
List of participants.....	18

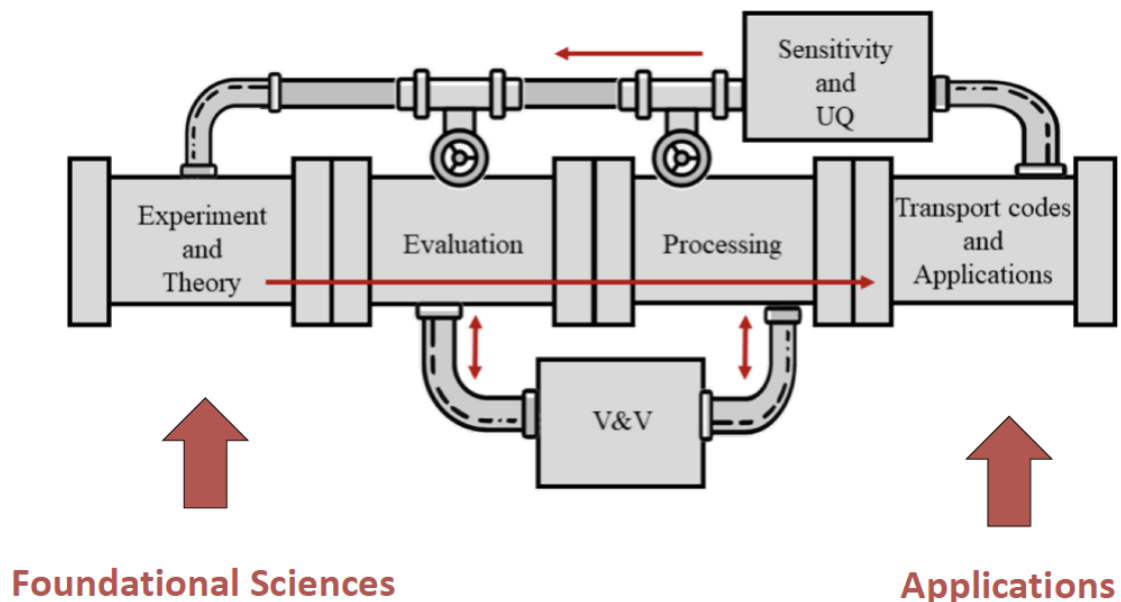
Objectives & white paper

The purpose of the event is to assess the actual capabilities, successfully deployed methods, tools and protocols, and future needs in terms of nuclear model code outputs in the General Nuclear Data Structure and their processing into useful applications forms. The event also focuses on building a modern durable partnership between fundamental sciences and application needs, in support of high-fidelity multi-physics simulation efforts.

The meeting brings together the different experts able to determine and support the necessary enhancements, processes and schemas, foreseen for the multiple applications in energy, non-energy, earth and life sciences to be served by GNDS and in support of high-fidelity Multiphysics simulation efforts. Eight experts attended (physically) the meeting.

Applications

The figure below shows how the nuclear data pipeline connects data from foundational sciences efforts to applications: energy, non-energy, space, medical, earth and life-sciences alike. Nuclear data evaluation and processing together play an integral role in this pipeline, and they are being modernized through the efforts of meeting attendees and others in the nuclear data community.



YANDF: An attempt to unify TALYS, EXFOR and ENDF data formats

A proposal is made to standardize the numerous TALYS [1] output files to a format which is close to an 'official' format, in this case YAML. The current version of TALYS is already rather systematic in terms of data files and meta data, and at least two codes, TEFAL and TAGNDS, successfully transform the TALYS output into nuclear data libraries. TENDL is based on this route. However, a higher degree of automation in the nuclear data pipeline and AI/ML applications require that consistent serialization of the TALYS output is taken a step further. Therefore, Yet Another Nuclear Data Format (YANDF) is presented here, primarily to provide a consistent schema for TALYS output enabling easy processing, not only by the TAGNDS and TEFAL authors, but also by others who want to automate processing of the TALYS results. In addition, starting from the fundamental definition of a nuclear reaction, the same schema can be used to provide the metadata for a large part of EXFOR and ENDF data. The resulting keyword-driven data files are not as non-descriptive as ENDF and not as heavy as EXFOR or GNDS, and should be an easy starting point for direct use, especially to numerically compare or plot data, or for parsing into GNDS, JSON, ENDF or other formats. The proposed format looks to be complete for the majority of reactions present in NDL's or EXFOR, though some extra metadata may need to be added for particular reaction such as break-up etc. A next step would be to provide a TALYS-reader which is capable of reading all output files and turn that into interesting summaries, statistics etc.

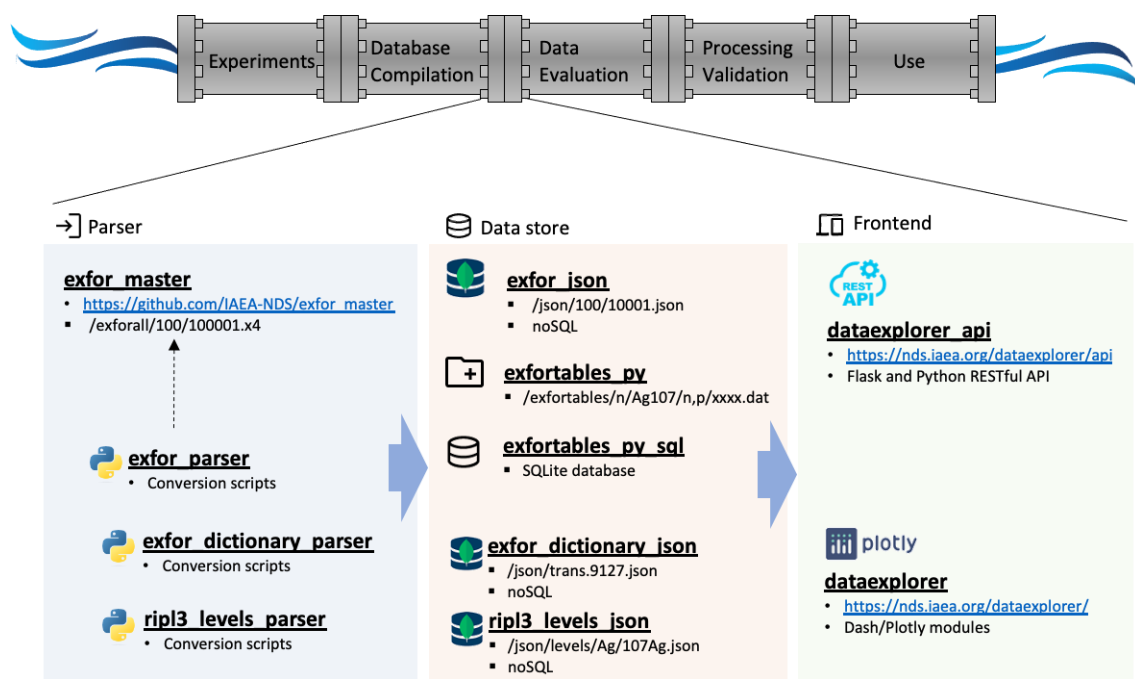
- make TALYS output more consistent. Simplify TALYS, EXFOR, ENDF data management with 'light' data exchange format.
- a question remains: what gets lost when storing data in ENDF-6 first? Test TAGNDS against TEFAL then endf2gnds
- Center Of Mass to Laboratory conversion is one-to-one, but inverse conversion is not always one-to-one.

Developing a Web Service for EXFOR Using RESTful API and JSON Schema

The Experimental Nuclear Reaction Database (EXFOR) is a collection of numerical data from the scientific literatures compiled by an international collaboration within the Nuclear Reaction Data Centre (NRDC). Despite the data being digitized and widely used for decades, the 80-column fixed width format with some tricks to reduce the data size, which originates from punch card legacy, is still in use. This format strongly restricts users from using EXFOR data directly with plotting packages and large-scale data analyses for modern AI/ML applications, such as developing outlier detection, substantially undermining the applications of modern techniques to nuclear data development. Solutions for the approaching of FAIR (Findable, Accessible, Interoperable, and Reusable) data and metadata in experimental nuclear data are currently lacking. The development of the nuclear data

pipeline is also experiencing stagnation in the evaluation stage where data retrieval from the EXFOR database is concerned.

As an initial attempt, we have developed a new web based nuclear data visualization tool [2] in 2020. This tool was built upon the reformatted EXFOR data in C5 by EXFORTABLES [1] and the pre-processed evaluated nuclear data libraries from ENDFTABLES [1]. Currently, in 2023, we are working on developing a new data flow to replace EXFORTABLES. The aim is to directly parse data from the original EXFOR database, allowing us to provide EXFOR data through a RESTful API and a new user interface. This development addresses the gap in the evaluation stage and partially aligns with the principles of FAIR data.



The conceptual representation of this approach is illustrated above. Our developments incorporate several key components: (1) preserving the EXFOR master files, (2) utilizing Python programs to parse and convert EXFOR and the EXFOR dictionary into JSON format, (3) incorporating a Python program to parse the RIPL3 discrete level data and decay data library in ENDF format and to convert them into JSON, (4) implementing a FLASK-based RESTful API, and (5) constructing a web interface using the Dash/Plotly framework, which are all open-sourced on GitHub [3].

- EXFOR supports folding data blocks (wrapping lines within tables). This is painful for end users. More generally speaking EXFOR is geared towards developers and exchange between data centers. Not meant to be user facing, so parsing is necessary.
- Computer format C4 was developed 30 years ago, C5 is a newer version.

- Several new interfaces for EXFOR are under development: X4i, EXFORTABLE, C5, X4+/-
- EXFOR has quality issues: all computer format translation of EXFOR must skip some entries that simply do not make sense, and unfortunately most compilers are no longer active.
- It is good to have several EXFOR parsers, same justification as having multiple processing codes.

TAGNDS and Autotalys update

TAGNDS was developed to enable producing evaluations directly in GNDS. The code is intended to simply be a translator, taking the outputs from TALYS and Autotalys and using them to instantiate FUDGE classes that can then be serialized to GNDS files. TAGNDS was recently updated to handle the updated outputs from TALYS 2.0 (with multiple incident energies in a single output file) and to work with the latest release of FUDGE (v6.2) [4]. It has been tested against multiple projectiles and targets and is consistently able to produce GNDS files. Some issues remain to be resolved, however: Autotalys outputs sometimes contain problems such as negative probabilities, values written with insufficient precision, and reactions with cross sections but no outgoing product distributions. TAGNDS reports these issues, and they should be fixed in Autotalys if possible.

GNDS-2.0 files produced by TAGNDS have been compared against files produced by TEFAL and the ENDF-GNDS translator. Results are mostly similar, but some remaining differences should be further investigated.

- TENDL is moving away from MF32 in favor of MF33. But the idea of storing MF32 as the 'original' evaluated with MF33 as 'eval2' style is possible in GNDS. Also, MF32 is necessary for doing UQ in self-shielding factors.
- As a recommendation for testing TAGNDS: in Autotalys input, mostly just use defaults except for setting 80 bins. Also consider using '-ntalys 100' to get 100 random files for generating covariances, although that takes a few hours. Make sure to remove '-thin' option. That should be equivalent to TENDL2023 generation.
- Should we add a threshold for ignoring reactions with very small cross section? Or should that be left for Autotalys? A similar question for very small Legendre coefficients.
- Some discussion about 'endfConversionFlags' update: we know default behavior if no endfConversionFlags supplied needs to be improved. It was suggested using MF=4 for elastic only, MF=6 for everything else except maybe photons (MF=12)
- TENDL authors are interested in storing MACS and reaction rates. TALYS can produce those, exemplified in the astrophysics section on the TENDL2023 [5]
- Some discussions occurred about best configuration management: should TAGNDS be packaged with FUDGE? With TALYS? lean for TALYS since reader depends on

TALYS output data form but still need a requirements.txt to point to specific FUDGE version

- Questions about adapting TAGNDS for other codes, in particular JAEA may be interested in getting CCONE -> GNDS translator working.

LLNL's FUDGE and GIDI+ Code Packages

FUDGE [4] and GIDI+[6] code efforts at Lawrence Livermore National Laboratory (LLNL) were presented. FUDGE is a Python based code package for reading, writing, modifying, processing and viewing nuclear data in the GNDS format. LLNL released its latest version of FUDGE (version 6.2) in May of 2023 at <https://GitHub.com/LLNL/fudge>. This version can be installed via "git clone/Makefile" or "pip install". Some changes to the FUDGE API have been made so some user scripts may need to be updated. FUDGE is developed as a toolkit for handling GNDS data; that is, users can write scripts in FUDGE to manipulate, view and process GNDS-2.0 as desired. The FUDGE team continues to develop scripts that are released with FUDGE that users may find useful. Several examples include:

- processProtare.py which processes a GNDS evaluated data file for Monte Carlo and deterministic transport
- checkGNDS.py which physics checks a GNDS evaluated data file.
- energySpectrum.py which makes outgoing energy spectra plots for an outgoing particle at a specified projectile energy.
- toACE.py which generates an ACE file from a Monte Carlo processed GNDS file. Currently only supports writing outgoing product distributions for neutrons.

GIDI+ is a C++ API for reading and sampling from GNDS data as needed by Monte Carlo and deterministic transport codes. GIDI+ is implemented in LLNL's Ardra and Mercury transport codes, and is currently being implemented into CERN GEANT4. When used for Monte Carlo transport codes, GIDI+ support running on GPUs. GIDI+ can be downloaded from <https://GitHub.com/LLNL/gidiplus>.

Recently, several features have been added to FUDGE and GIDI+ to decrease the access time for reading data from a file. These include, hybrid XML/HDF5 file support, lazy parsing of data containers (ergo, each data container is only parsed into its GIDI+ class if needed) and pre-summed multi-group data.

The EMU (Evaluated Means and Uncertainties) code being developed at LLNL by Kyle Wendt. EMU creates realizations of an evaluated data file from its mean and covariance data. EMU uses FUDGE to read, manipulate, write and process GNDS files. It is expected EMU to be released soon on GitHub. Finally, several demonstrations of the FUDGE scripts were made.

- The CERN FLUKA team should be made aware of GNDS, as GEANT4
- JAEA recommendation on energySpectrum.py plotting is to include the ability to plot grouped data: cross sections, spectra, and transfer matrices at least, maybe average energies too.
- JENDL-5 website has an interface for looking at various quantities in a convenient 70 groups structure with $1/v$ flux weighting.

The FUDGE translation of the TENDL-2021 library from ENDF-6 to GNDS

The conversion of TENDL libraries from ENDF-6 to GNDS is an indirect approach to making the output from the TALYS nuclear model code system available in the new Generalized Nuclear Data Structure (GNDS) format. The LLNL processing code FUDGE was used to translate TENDL-2021 from ENDF-6 to GNDS. An overview of this work is provided, and the different warning and error messages are discussed.

- There are some issues remaining in translating TENDL-2021: many are related to MF=31 MT=452 (total nubar) covariances.
- During the making of TENDL-2023, the IAEA and LLNL want to ensure files translate cleanly using the latest Autotalys settings and hopefully reproduce any issues.
- Further translation testing should include JENDL-5 Errata [7]. As for the decay updates it already fixes around half of the FUDGE complains about the use of the 'NT=4' flag. Based on the ENDF-6 manual, that flag should likely be set to NT=6. More information available at https://www.ndc.jaea.go.jp/jendl/j5/JENDL-5_Errata.html
- The above clears up some errors, but 43 TENDL-2021 translation errors still remaining for the neutron sub-library.
- Looked in-depth at the JENDL-5 n + d evaluation: for MT=16 (n + d -> 2n + p), FUDGE complains because a total of 4 products are listed:
 - n with multiplicity=2 up to 20 MeV, 0 above with full distribution
 - n with multiplicity = 0 up to 20 MeV, 2 above, with no distribution
 - Same for p: broken up into two separate products at 20 MeV. Is that legal ENDF-6? How do other processing codes handle this data?
- Some other translation issues revealed problems in the FUDGE translation tool that should be fixed by FUDGE developers.

Handling of GNDS in FRENDY and JAEA recent activity

FRENDY version 2.01 was released in November 2022 [9]. The manual of FRENDY version 2 was published as a JAEA report (JAEA-Data/Code 2022-009) in February 2023. These can be freely downloaded from the JAEA website [8].

JAEA use the GIDI+ of LLNL to handle the GNDS format file in FRENDY. FRENDY uses the NuclearDataObject module to process a nuclear data file. JAEA is developing the converter module to copy the data from GIDI+ to the NuclearDataObject module. GIDI+ is an excellent tool to treat the GNDS format file. The FRENDY developer requested the implementation of the resonance parameter handling function in GIDI+ to process the ACE and multi-group cross section library in FRENDY. GNDS handling function in FRENDY will verify with the comparison of the processing results of ENDF-6 format files and GNDS format files after completing the development of the converter modules.

JAEA group initiated the development of a high-fidelity multi-physics simulation platform JAMPAN (JAEA Advanced Multi-Physics Analysis platform for nuclear systems). The current target of the multi-physics simulation is the coupling of the neutronics and thermal-hydraulics calculations. Coupling with other calculations such as a structure calculation is a future work. Users can easily change calculation codes in JAMPAN. To reduce the reliance on calculation codes, JAMPAN converts calculation results into the JAMPAN data file. Other calculation codes can be added by the development of the input generator and output reader modules. The JAMPAN data file is the HDF5 format file. JAEA uses the GNDS format manual as a good reference for designing the JAMPAN data file format.

- GNDS in FRENDY is using GIDI+ as interface for both reading and writing GNDS.
- There is now a need to add resonance parameter reading and writing support to GIDI+

NJOY2016 modernisation and GNDS support in scion

The NJOY modernisation strategy [10] in which the required format components (for ENDF and ACE) were developed first is moving to the next stage in which development of processing components has started. The first among there is scion, modern NJOY's component for the interpretation, linearisation, integration and differentiation. The current state of this new component already allows us to perform some diagnostics on the methods used in NJOY2016 for the processing of ACE files. In addition, to providing the basic operations that will be used in various processing modules, scion will also provide a format agnostic data interface capable of reading in and writing out data from various evaluated nuclear data format including ENDF and GNDS.

- ENDFtk and ACEtk [9] meant to provide low-level access layers for different formats. ENDFtk is complete, ACEtk complete for most types of ACE data. GNDStk also under development but still needs work.
- Scion [9] algorithms are showing big improvement over NumPy on timing tests for polynomial series operations (root finding, etc.). Some discussion about trying Chebyshev polynomial as approximation to functions.
- Scion also has Python interface using PyBind11. Comment about PyBind11: very useful but must be a bit careful not to do extra copies on return.
- Detailed demo of ACEtk and ENDFtk (also using PyBind11). Some interesting demos of linearization, it was shown that sometimes finding maximum difference between polynomial and linearized version can give noticeably smaller grid than linearizing by bisection.

<https://GitHub.com/njoy/scion/blob/feature/examples-1/python/examples/h1-elastic-angular-distribution.ipynb>

<https://GitHub.com/njoy/scion/blob/feature/examples-1/python/examples/h1-capture-linearisation.ipynb>

Recommendations

- GNDS-2.0 format specification has been finalized and will soon be published by the NEA (expected mid 2023) https://www.oecd-nea.org/jcms/pl_39689/specifications-for-the-generalised-nuclear-database-structure-gnds
- FUDGE-6 has been demonstrated to rigorously test the quality of raw ENDF-6 libraries in the course of translating to GNDS-2.0 and running physics checking. It is recommended to use this capability to translate and test recent and in-development library releases and report issues back to library developers and maintainers.
- A short-term goal is to make GNDS-2.0 translations of several libraries available to enable processing codes and users to test more extensively. Additional benefit is ensuring clean translation to GNDS also improves original library quality.
 - ENDF-VIII.0, ENDF-VIII.1 (to be released in 2024)
 - JENDL-5
 - TENDL-2023 (to be released in 2023)
 - JEFF-4 (to be released in 2024)
- The capability to directly produce GNDS-2.0 evaluations is of growing importance as evaluators wish to add physics which GNDS supports but ENDF-6 does not. The TAGNDS code implements the ability to translate the output of the TALYS Hauser-Feshbach and the Autotalys (TALYS + resonances + covariances) nuclear data evaluation codes directly into GNDS-2.0. It is recommended to integrate TAGNDS into Autotalys and expand its testing by exercising it as part of TENDL-2023 production.
- TAGNDS opens up a new route for producing GNDS-2.0 evaluations with physics not supported by the ENDF-6 format. It is recommended to test TAGNDS against the indirect route of first producing ENDF-6 and then translating to GNDS, to ensure TAGNDS agrees with or improves upon the indirect route.
- Infrastructure for processing and using GNDS nuclear data is expanding, with multiple institutions updating their codes to handle GNDS-2.0. Application needs are already encouraging these efforts to be expanded. Processing results should be compared against results from processing ENDF-6, and findings should be shared amongst institutions.
- Codes for checking and validating nuclear data must also be updated to support GNDS. These should include plotting and comparison tools as well as physics checkers and format validators. Interactive tools such as Python scripts, Jupyter notebooks, matplotlib and Plotly are preferred over legacy 'input deck'-driven plotters and checkers. Recipes for checking and generating plots should be shared.
- It is noted that multiple new codes, numerical recipe, protocol are being developed using Python, C and C++ and released as open source. These are facilitating greater cooperation amongst nuclear data evaluators, processors, and users. The GNDS-2.0

standard is expected to work more seamlessly with these new codes and streamline the road from improved science to applications. The expanded use of these modern tools should help attract new talent to the field.

- It is noted that providing numerical recipe, scripted examples and protocols using FUDGE, GNDS or scion support dissemination and wide-spread use of nuclear data.
- It is also noted that just as GNDS is modernizing how evaluated nuclear reaction data are produced and used, other nuclear data efforts (such as nuclear structure, modeling, and experimental data compilation teams) are also working to modernize their formats and toolkits. These efforts are expected to work in synergy with the ongoing effort to improve GNDS.

References

- [1] TALYS, EXFORTABLES <https://www-nds.iaea.org/talys/>
- [2] <https://nds.iaea.org/dataexplorer/>
- [3] <https://GitHub.com/IAEA-NDS/>
- [4] FUDGE <https://GitHub.com/LLNL/fudge>
- [5] TENDL-2023 https://tendl.web.psi.ch/tendl_2023/tendl2023.html
- [6] Gidi+ <https://GitHub.com/LLNL/gidiplus>
- [7] JENDL-5 <https://wwwndc.jaea.go.jp/jendl/j5/j5.html>
- [8] XML Schema for GNDS-1.9: <https://www.oecdnea.org/download/wpec/gnds/gnds.xsd>
- [9] FRENDY https://rpg.jaea.go.jp/main/en/program_frendy/
- [10] NJOY2016, ACEtk, ENDFtk, GNDStk, scion <https://GitHub.com/njoy/>

List of participants

S. No.	Authority	Personal Details
1	JAPAN	Mr Kenichi TADA 4-49 Muramatsu, Tokai, Naka District, Ibaraki Prefecture 319-1112, TOKAI Email: tada.kenichi@jaea.go.jp
2	UNITED STATES OF AMERICA	Mr Bret BECK Lawrence Livermore National Laboratory 7000 East Avenue LIVERMORE, CA Email: beck6@llnl.gov
3	UNITED STATES OF AMERICA	Mr Gert GODFREE Lawrence Livermore National Laboratory P.O. Box 808 LIVERMORE, CA Email: gert1@llnl.gov
4	UNITED STATES OF AMERICA	Mr Wim HAECK Los Alamos National Laboratory P.O. Box 1663 LOS ALAMOS, NM Email: wim@lanl.gov
5	UNITED STATES OF AMERICA	Mr Caleb MATTOON Lawrence Livermore National Laboratory 7000 East Avenue, L-414 LIVERMORE, CA 94550 Email: mattoon1@llnl.gov
6	IAEA	Jean-Christophe SUBLET Scientific Secretary Head-Nuclear Data Service Unit, Nuclear Data Section, IAEA P.O. Box 200 1400 VIENNA Email: j.c.sublet@iaea.org
7	IAEA	Arjan Koning Head-Nuclear Data Section, IAEA P.O. Box 200 1400 VIENNA Email: a.koning@iaea.org
8	IAEA	Shin Okumura Nuclear Data Service Unit, Nuclear Data Section, IAEA P.O. Box 200 1400 VIENNA Email: s.okumura@iaea.org

Nuclear Data Section
International Atomic Energy Agency
Vienna International Centre, P.O. Box 100
A-1400 Vienna, Austria

E-mail: nds.contact-point@iaea.org
Fax: (43-1) 26007
Telephone: (43-1) 2600 21725
Web: <http://www-nds.iaea.org>
