The ACTS Center for Software Sustainability

FINAL REPORT
SEPTEMBER 2011

Dr. Leroy A. Drummond (PI)
Lawrence Berkeley National Laboratory
One Cyclotron Road, MS 50F-1650
Berkeley, CA 94720-8139
LADrummond@lbl.gov

1. Introduction

This report covers the activities carried out by the project *The ACTS Center for Software Sustainability*, whose proposal was submitted to ASCR in 2007. These activities have their roots in the project *Advanced Computational Testing Software* (*ACTS*) *Toolkit*, which was part of the DOE2000 Program. Driven by the state of the software, the technology and the nature of the tools in the Toolkit, the project was reformulated in a project that better represented the goals and activities within ACTS, *An Expanded Framework for the Advanced CompuTational Software* (*ACTS*) *Collection*. At the heart of this project was the consolidation of lessons learned and demands from the computational science community that guided us to the creation of the now well-known DOE ACTS Collection. The goal of the project was to increase the pay-off of DOE ASCR investment in the development of the general purpose tools in the DOE ACTS Collection, reduce the duplication of efforts, and minimize the development time of high-end computer applications while maximizing the life of the codes.

Over time, and responding to ASCR's guidance, a reformulation of strategies and goals was implemented, together with additional services, to promptly respond to advances in hardware, software development practices, evolution of tools into toolkits, and problems in computational sciences (see Figure 1). In addition, and in direct support of our activities, a number of reports from the engineering and scientific communities have unanimously emphasized that mechanisms to reduce the effort required for software development, testing and evolution are imperative for improving the nation's high-end computing capabilities, and for achieving an optimal use of the available computing resources. These mechanisms are particularly important for applications that require large-scale computer simulations, and also because the software functionality lifespan is usually much longer than the lifespan of the hardware they run on.

ACTS stems from the realization that the work required for the development of scientific application codes for high-end architectures has increased with computer power; achieving high performance on these architectures usually requires a significant effort and the implementation of very specialized kernels. To address this issue, ACTS has built a state-of-the-art software infrastructure to facilitate the evolution and adoption of robust tools in response to changes in user needs and computer technology. ACTS focuses on: a solid base of tools (the DOE ACTS Collection), high-quality software certification, inclusion of new solutions to computational problems, collaboration with DOE computer facilities and other software initiatives, encouragement of feedback from users, expertise tracking, increased visibility of the ACTS Information Center, and dissemination activities.

This report describes our efforts towards improving and building mechanisms that have facilitated the adoption of robust and high performing software tools by the scientific community

Figure 1 - Evolution of the DOE ACTS Collection Project

at large, our overarching efforts, and our search for opportunities where robust, high performing tools can be effectively employed. Table 1 summarizes the functionalities of the tools that currently form the core of the DOE ACTS Collection.

In general, computer facilities provide a variety of third-party software, which may include libraries for (dense and sparse) linear algebra calculations, METIS (for graph partitioning), FFTW (a library for computing the Discrete Fourier Transform in one or more dimensions), and NAG (a collection of mathematics and statistics libraries). More specialized software may include, among others, AMBER, CHARMM, GAMESS, GAUSSIAN, LAMMPS, MOLPRO, NWChem and VASP (for molecular dynamics and quantum chemistry), ABINIT (for the total energy, charge density and electronic structure calculations), and Abaqus, Ansys, and LS-DYNA (general purpose finite element analysis packages). Noteworthy, some of this specialized software may also invoke functionalities provided by tools in the ACTS Collection. As examples, NWChem is built atop Global Arrays, and VASP can be configured to use ScaLAPACK for matrix factorizations and diagonalizations. ScaLAPACK is also at the heart of the Microwave Background Analysis tool, MADCAP. Additionally, the tools listed in Table 1 can be used in different scenarios, for example for code development, prototyping, and large simulations that have the potential to lead to scientific breakthroughs.

Basically, we have expanded our outreach and dissemination of ACTS tools, and high-level user support to application developers while writing their codes and selecting functionalities provided by the tools. Figure 2 presents the scope of our work in the ACTS Center, which we group in four main areas: long-term maintenance, independent testing and evaluation, outreach and dissemination and high-level user support. In the sequence, we describe the goals of each area, motivation and corresponding accomplishments.

Table 1 - Tools that form the "solid base" of the DOE ACTS Collection

Tool	Functionalities and Status
ATLAS	Automatic tuning of basic linear algebra subroutines. Developed at the University of Tennessee, Knoxville
Aztec (AztecOO)	Algorithms (based on Krylov subspaces) for the iterative solution of large sparse linear systems. Developed at Sandia National Laboratories.
Hypre	Algorithms (based on Krylov subspaces) for the iterative solution of large sparse linear systems, intuitive grid-centric interfaces, and dynamic configuration of parameters. Developed at Lawrence Livermore National Laboratory.
Global Arrays	Library for writing parallel programs that use large arrays distributed across processing nodes; "shared-memory" programming interface for distributed-memory computers. Developed at Pacific Northwest National Laboratory.
Overture	Library for solving computational fluid dynamics and combustion problems in complex moving geometries
PETSc	Tools for the solution of PDE problems that require solving large-scale, sparse linear and nonlinear systems of equations. Developed at Argonne National Laboratory.
SUNDIALS	Solvers for large systems of ordinary differential equations, nonlinear algebraic equations, differential-algebraic equations, and sensitivity analysis. Developed at Lawrence Livermore National Laboratory.
ScaLAPACK	Library of high performance dense linear algebra routines for distributed-memory architectures. Developed at the University of Tennessee, Knoxville, and UC Berkeley.
SLEPc	Software package built on top of PETSc for the solution of large sparse eigenvalue problems. Developed at the Universidad Politecnica de Valencia, Spain.
SuperLU	General-purpose library for the direct solution of large, sparse, systems of linear equations. Developed at UC Berkeley and Lawrence Berkeley National Laboratory.
TAO	Tools for the solution of large-scale optimization problems, including nonlinear least squares, unconstrained minimization, bound constrained optimization, and general nonlinear optimization. Developed at Argonne National Laboratory.
TAU	Tools for analyzing the performance of programs written in C, C++, Fortran or Java. Developed at the University of Oregon and Los Alamos National Laboratory.

2. A Comprehensive Solution to Long Term Software Sustainability

Following up on the previous two phases of the DOE ACTS Collection project and having identified an already robust set of general purpose tools and their effectiveness to enable science, the current project has focused on the sustainability of the tools. Given the number of DOE scientific applications that depend on the availability of the ACTS tools, software sustainability has thus become a key priority for the project. We look at several aspects of the software sustainability: tool availability in emerging platforms, tool performance and scalability, and the tool ability to address pressing HPC challenges. As illustrated in Figure 2, our thorough response plan to address sustainability has four major components that have spun synergetic activities within the project: Long-Term Maintenance, High-Level User Support, Independent Testing and Evaluation and Outreach and Dissemination.

2.1. Long-Term Maintenance

Goals: Make software available beyond its original phase of development (and funding), and at the same time facilitate software evolution. Ascertain back and forward compatibilities between a software tool and all other software tools and applications that use it (enhancements and optimizations to a software tool may result in a different version of the tool).

Figure 2 - Overview of the functionalities in the ACTS center and its thorough plan for sustaining a robust set of computational tools.

Motivation: A number of application codes rely on functionalities that are available in the current set of tools in the ACTS Collection. Therefore, these codes need to have the tools already in place on a platform before application developers start the migration process of their software, and the tools need to be available throughout the evolution of the application.

Accomplishments: Development of a software dependency graph that defines a list of prerequisites that is necessary for a tool or set of tools to work on a given platform. An example of this dependency information is listed in Table 2. For instance, ScaLAPACK depends on the BLACS (Basic Linear Algebra Subroutines) and PBLAS (Parallel BLAS), PETSc depends on BLAS and LAPACK (Linear Algebra PACKage), and SLEPc and TAO depend on PETSc. All the tools in the DOE ACTS Collection are parallel tools and depend on a messaging passing library like MPI. Also, for purposes of sustainability, the ACTS tools depend on other system support tools like C, C++ and Fortran compilers. Using auto-tuning tools like ATLAS, we can obtain optimized versions of the BLAS and some subroutines from LAPACK. The optimized versions are then used by all of the numerical tools in the DOE ACTS Collection. The software dependency graph can map the performance of a tool and its software dependency tool chain to sustain a given performance level. For instance a tool that depends on BLAS may have more than one option of the BLAS libraries in the system, all of the BLAS library versions may lead to different performances.

We collaborated early on this project with the numerical tool group at CRAY Inc. and engaged in discussions about performance related issues and development for the numerical tools in ACTS. Now, CRAY provides some versions of these tools with the LibSci scientific library but usually its releases are behind the developer versions of the library. Nevertheless, we see this

Table 2 - Tools in the ACTS Collection and their dependencies. The functionalities provided by the tools are summarized in Table 1.

Tool	BLAS	LAPACK	MPI	Threads	Python	BLACS	PBLAS	Math Lib	Compiler utilities	Autoconf	Other Tools
ATLAS	X	X						X	C, CPP		
Trilinos	X	X	X		X			X	C, C++, CPP	X	Trilinos core functionality
Hypre	X	X	X					X	C, CPP	X	
Global Arrays	X		X	X				X	C, CPP		
OPT++	X	X						X	C++, CPP		
PETSc	X	X	X		X			X	C, C++, CPP, Fortran		
SUNDIALS	X	X	X					X	C	X	
ScaLAPACK	X	X	X			X	X	X	C, Fortran	X	
SLEPc	X	X	X					X	C, Fortran	X	PETSc
SuperLU	X	X	X					X	C, Fortran	X	
TAO	X	X	X					X	C++	X	PETSc
OVERTURE	X	X	X					X	C, C++	X	
CUMULVS			X						C		PVM
TAU									C, C++, Fortran	X	

as a very successful interaction for ACTS in effectively sharing expertise and capability with computer vendors to extend the acceptance and life of tools in the DOE ACTS collection.

2.2. High-Level User Support

Goals: Deployment of an extended level of support for all ACTS tools, targeting different levels of user expertise and familiarity with a given tool, in collaboration with tool developers and staff at computing centers.

Motivation: The state of user support (e.g. documentation) and portability (e.g. easiness of installation) of ACTS tools vary, but both of these aspects are important for the sustainable and long-term maintenance of the tools. Therefore, the project sought to complement the developers' support, to achieve a homogeneous level of support for all tools, see Figure 3.

Accomplishments: A successful collaboration with the NERSC User Support Group for which the ACTS team has been added to the NERSC Trouble Ticketing System. In this way, ACTS members can interact with users on issues ranging from tool selection, tool use to solving coding problems and bugs with the tools. ACTS provides also user support through a more general email list acts-support@nersc.com (175 requests from Sept 2006 to August 2011).

One key aspect when maintaining and supporting research tools that do not have a well designed distribution infrastructure is support at installation time. Basically, if a user cannot install the tool on her or his platform after a couple of hours the user stops trying. We have collaborated with ParaTools Inc., in the creation of live DVDs of the DOE ACTS Collection. The tools are preinstalled and include other support tools like debuggers, MPI and other dependent software.

Figure 3. ACTS complements support and portability to users beyond the tool developers' basic distributions of the software.

In collaboration with the University of Alicante in Spain, we have developed PyACTS, a python based user interface to some tools in the ACTS Collection. For tools like PETSc and Trilinos (where AztecOO is now hosted) we have integrated third party python interfaces with PyACTS.

ACTS News is a quarterly newsletter sent to know users of ACTS tools. Members are added to this list after their participation at one of the DOE ACTS Collection workshops and short courses or upon requests received at support@acts.nersc.gov.

ACTS has also provided better understanding of the differences between tools that provide overlapping functionalities, to guide users through the tool selection process. A centerpiece of the high-level user support is the newly design ACTS Information system, acts.nersc.gov, that now contains extensive examples on the use of the tools and performance graphs.

As an example of the metrics of success of activities in 2.1 and 2.2, a NERSC Principal Investigator wrote on his NERSC account allocation renewal:

> *Just a comment about why we use NERSC almost exclusively. Our work at NERSC is strongly dependent on PETSC and SLEPc [tools in the DOE ACTS Collection] in their implementations for intrinsically complex matrices. The NERSC implementations generally work and the support we get when they do not is simply superb. For us these services are the difference between publishing a dozen papers a year (including Physical Review Letters and Science) and getting nothing done.*

2.3. Independent Testing and Evaluation

Goals: Creation of an independent set of verification mechanisms to improve tool migration to new system environments and platforms, to ensure proper installation and also to obtain preliminary performance measurements on new platforms.

Motivation: Installation, testing and evaluation are integral tasks in software development projects because software becomes more mature and robust through progressions of its

Figure 4. Areas of expertise of participants in the ACTS Collection Workshops held at LBNL (starting in 2000). The work of about 25% of the participants relates to applied mathematics and scientific computing; while about 18% of the participants work on a large variety of problems (indicated by *other* in the chart). The figure refers to an audience comprised of 53% of students, 15% postdoctoral fellows, 22% researchers and faculty, and 9% representatives from industry, supercomputer centers, computer vendors, ISVs, and the military

versions and interactions with its users. Independent tests and evaluations provide tool developers with feedback on the implemented functionalities and unforeseen problems on particular system environments and platforms.

Accomplishments: Overall, we have installed and maintained ACTS tools with a sustainable performance on NERSC platforms and through migrations from IBM SP systems, Linux-based Clusters to the CRAY XT and CRAY XE systems. One of the main successes of this activity has been the ability to deal with multiple versions and the software and their interoperability within the DOE ACTS Collection. While tools like PETSc have a build system that enables users to automatically install third party software that PETSc can call or interoperate with, for a larger user base a more sustainable approach is needed. In the ACTS framework with install all the tools individually to ensure that we have the most robust versions of the tools installed. In some instances this has involved solving interoperability issues that have been unknown to the individual tool developers. ACTS has been effective by bringing the combine functionality to the hands of DOE application developers.

2.4. Outreach and Dissemination

Goals: Ensure that all ACTS tools are accompanied by examples that illustrate different functionalities and reflect distinct levels of expertise. Create training opportunities, through tutorials and hands-on sessions, to computational scientists from academia, laboratories, and industry. Provide references to information available in other places, for example to tools that are not part of the collection but that may offer alternative functionalities to specific applications.

Motivation: Training opportunities focusing on ACTS tools expose students to technologies that are usually not presented in regular courses, and simultaneously provide tools developers with important feedback about the tools.

Accomplishments: The tutorials and hands-on sessions organized by the project have been attended by hundreds of students and computational scientists; see Figure 4 for a view of the

Table 3. List of Collaborations

wide spectrum of application scientists. In some cases, participants in the training events have later contributed to further development of the tools. Since 2007 we have organized 12 short lectures and courses on the DOE ACTS collection, have given 3 invited keynote talks at international conferences, have given yearly full day tutorial at the DOE SciDAC meetings, and have been invited to numerous events to present the ACTS tools. All of these activities in addition to organizing the 12th DOE ACTS Collection workshop in August 2011, which brought the total of ACTS alumna to over 500.

3. Infrastructure of the ACTS Center: Collaborations and Interactions

The creation of a software sustainability center that provides a high level of support to users and tool developers has been promptly addressed by the ACTS project. Over the years, we have managed to bridge the gap between tool and application developers that in many cases leads to duplication of efforts. Our efforts have allowed for an ampler software development, whereby common issues like well-documented interfaces, distribution, upgrades, optimization, tuning and debugging have been effectively and uniformly addressed. We have addressed elements of the software sustainability center that relate to the lifecycle of the software and the future of high-end hardware, which are the core infrastructure for the ACTS Center and collaborations, see Table 3. We have defined as collaborations the agents with whom we have worked directly in the implementation of this infrastructure. Our interactions with other centers were geared to support their users and software and could have become direct collaborations with the ACTS Center if funding had become available.

The DOE ACTS Collection has been a dynamic set of tools, ready to address issues related to tool utilization, evolution, superseding, and integration of new technologies. This dynamic aspect accommodates changes to hardware and software driven by the needs of the computational science community.

The element of the Center that deals with tool evolution and integration of new functionalities has also addressed the increasing demand for tool interoperability and more uniformity in the documentation and integrated user interfaces. Tool developers have been informed of the interoperability dependencies between their particular tools and the way we have resolved some incompatibilities between interfaces. Thus, we have facilitated the integration of new functionalities and support of these in existing tools.

Furthermore, we have been working on PyACTS, a tool written in Python that provides access to functionalities in the ACTS Collection. Currently, PyACTS users can access the full functionality available in ScaLAPACK.

Our testing and independent evaluations has benefited from our direct collaborations with DOE high performance computing centers like NERSC, and computer vendors, like CRAY Inc. These collaborations have provided the required access to the platforms in order for us to perform adequate testing that is relevant to the tools before the acceptance phase of a computer is cleared. In turn, our early access testing will also enable a more reliable and correct installation of the tools and all the application software that depend or will depend on the ACTS tools. The above activities make having active collaborations with the DOE leadership and flagship facilities, and the DOE outreach centers all the more important.

We have sought to continue international collaborations and interactions, through short courses, journal publications, and organization of conferences and workshops. These collaborations and interactions have given greater visibility to the different phases of the ACTS Project. They have also generated interest in the tools in the ACTS Collection in a wider community, and placed the tools in a world-leading class. The more users of the tools the more robust and better equipped the tools become to face emerging challenges in computational sciences and cope with software platforms.

4. Comments and Conclusions

We have been mostly working with the DOE flagship computing facility, NERSC, and the ACTS Center infrastructure has supported and helped NERSC users and collaborated with the user support staff. These interactions have enabled us to pass expertise on the ACTS tools and related HPC technologies to the NERSC consultant and effectively speed up the development of High End codes. This will become even more relevant as the HPC technologies continue to gear up for large computational scales and we anticipate even more users deploying ACTS tool. Ultimately, these services need to be expanded to other DOE computing facilities, in the case that financial support for these activities becomes available. The need for ACTS support at these facilities has already been pointed out by more than one of the application development team that use NERSC and other DOE supercomputer facilities. We have initiated talks with some members at ORNL and ANL but ultimately we need to add personnel from those facilities into the project to effectively interact with those facilities.

While implementing the original agenda for this project, we had to revise a few of the following tasks:

Table 4 - Tools in consideration for inclusion to DOE ACTS Collection

Tool	BLAS	LAPACK	MPI	Threads	Python	BLACS	PBLAS	Math Lib
OSKI	X	X						X
Zoltan			X					
ML	X	X	X					X
BELOS	X	X	X	X				X

- **Software distribution plans changed**. Changes in the software technology have now provided more robust configuration and build mechanisms and all tools in the ACTS Collection are now using either **autoconf** or **cmake.** Both software utilities are freely available and their functionalities have also enabled use to investigate the integration of auto-tuning tools to these utilities. In addition, one of our lessons learned in this processes is that user preferred to use ready-to-use tools rather than spending 5 or more hours installing a tool. Our solution was to provide users with LiveDVDs that contained preinstalled versions of the libraries and emulate a Linux environment that runs in almost all available platforms today. Specifically, we have tried the LiveDVDs on Linux based systems, Windows and Mac OS.

- **Integrated forums with high performance computer and software vendors, computer centers, and domain experts.** From interactions with our collaborators, the best format for hosting these forums have been at already schedule international conferences because at the same time this funding cycle has coincide with many other strategic meetings organize by DOE and other national and international agencies with relevance to software for extreme scale. Thus, we have effectively interacted with the community instead through our organized mini-symposia, workshops and by taking a leadership role in events organized by different funding agencies.

- **Addition and retirement of tools in the collection.** In this funding cycle we have retired 4 tools from the collection. Primarily because these tools are not in demand any more as the technological changes have imposed different a different set of demands. The functionality not longer supported by the retired tools is now being either supported by another ACTS tool or users have been provided with references to a different tool not in the collection. Based on user feedback and need to better address the multicore and many core technologies, we have been installing, deploying and testing other tools which are being consider for a formal inclusion in the ACTS collection (see Table 4)

The multicore and manycore challenges have now posed a different problem for the entire HPC software stack. ACTS has taken a leading role in providing an infrastructure that supports users of ACTS tools without imposing major changes to the user interfaces. We have launched a parametric research approach that enables the automatic tuning and optimization of the application codes that use ACTS tools. Figure 5 summarizes our approach that currently includes the creation of parameters that help with the tool installations, software dependencies, algorithmic tuning and optimization and finally deployment of the ACTS tools in multicore platforms

5. List of Deliverables

Tables 5a and 5b summarize all our deliverables in the different functional areas of the ACTS project.

6. References

- L. Anthony Drummond, Vicente Galiano, Violeta Migallón, José Penadés: Interfaces for parallel numerical linear algebra libraries in high level languages. Advances in Engineering Software 40(8): 652-658 (2009)

- L. Anthony Drummond, Vicente Galiano, Violeta Migallón, José Penadés: PyACTS: A Python Based Interface to ACTS Tools and Parallel Scientific Applications. International Journal of Parallel Programming 37(1): 58-77 (2009)

- Drummond, L.A.; Galiano, V.; Migallón, V.; Penadés, J. "PyACTS: A Python based interface to ACTS tools and parallel scientific applications". International Journal of Parallel Programming. Vol. 10.1007/s1076. (2008)

- L. A. Drummond, V. Galiano, Violeta Migallón and José Penadés. "PyACTS: A High-Level Framework for Fast Development of High Performance Applications". Lecture Notes in Computer Science. Vol. 4395. (2007) 417-425

- L. A. Drummond, V. Galiano, V. Migallón and J. Penadés, *High-Level User Interfaces for the DOE ACTS Collection*, Lecture Notes in Computer Science 4699, Proc. Of the PARA 06 Workshop (revised selected papers), pp. 251-259, Springer-Verlag, 2007.

- O. Marques and P. J. Vasconcelos, *Evaluation of Linear Solvers for Astrophysics Transfer Problems*, Lecture Notes in Computer Science Vol. 4395, Proc. of the VECPAR'06 Conference, pp. 466-475, Springer-Verlag, 2007.

- O. Marques and T. Drummond, IJHPCA Special Issue on Tools in the ACTS Collection, Preface, Report LBNL-61557, 2006.

- Drummond, O. Marques, J. Roman and V. Vidal, *A Survey of High-Quality Computational Libraries and Their Impact in Science and Engineering Applications*, Lecture Notes in Computer Science, Vol. 3402, Springer-Verlag, Berlin, 2005.

- O. Marques and T. Drummond, *Building a Software Infrastructure for Computational Science Applications: Lessons and Solutions*, in Proc. of the Second International Workshop on Software Engineering for High Performance Computing System Applications, St. Louis, MO, May 15, 2005.

- T. Drummond and O. Marques, *The Advanced CompuTational Software (ACTS) Collection*, ACM TOMS, 31:282-301, 2005.

Figure 5. ACTS strategy for sustainable performance on multicore systems.

Table 5.a – Services for long-term software maintenance, independent test and evaluation.

Elements of the ACTS Center		Summary of Deliverables and Accomplishments
Long-Term Software Maintenance	Gear up application developments on next generation of computer systems and beyond	• Has maintained a collection of robust software tools for high-end computing and that implement numerical algorithms, support for code development, run time support and visualization, parallel debugging, data management, and automatic tuning. • Has maintained a list of critical tools in the collection based on application demands and supported installation at NERSC and answered user questions through acts-support @NERSC • Has retired tools that no longer matched the technology; CUMULVS and Opt++. • Has installed and supported ACTS tools at DOE flagship, have initiated contacts with personnel at the DOE leadership facilities at ANL and ORNL. • Has established and maintained ongoing collaborations with CRAY Inc. • Have designed and implemented the ACTS tool dependency graph. • Has been maintaining of tool dependency graph. • Have began to develop a parametric approach to support and auto-tune tools for multicore systems.
Independent Testing and Evaluation	Software testing and evaluation	• Produced tool dependency graph. • Created and updated a test suite per tool in the collection and dependable software. • Have participated in NERSC procurement by providing early installation and testing of a critical subset of tools. • Feedback to computer vendors. • Have performed independent tool evaluations: installations, documentations, performance and functionality.

Table 5.b – Services for outreach and dissemination, and high-level user support.

Elements of the ACTS Center		Summary of Deliverables and Accomplishments
Outreach and Dissemination	Training and dissemination of best practices	• Have organized 12 yearly DOE ACTS Collection Workshops in Berkeley • New ACTS Information Center with examples of tool use and performance. • Have been providing tutorials, short courses and lectures on-line. • Have produced electronic newsletters and information on ACTS Tools and related computational science activities. • Have periodically surveyed the community on software use and needs. • Have gathered feedback from the community on the software tools and the ACTS Center.
High-Level User Support	Consulting for users and high-end computing centers	• Has provided High-level consulting on ACTS tools: tool selection and utilization through NERSC trouble ticket system and acts-support@nersc.gov • Have maintained an acceptable response time to queries from users. • Have actively promoted ACTS software beyond DOE computational facilities through lectures at the national and international level. • Have kept information to consistently support tool interoperability and possible interoperable interfaces. These have been communicated to the relevant tool development teams • Produced 6 versions of ACTS LiveDVDs with pre-installed versions of ACTS tools. Versions were updated every 6 months as new versions of the libraries became available, and distributed primarily during the DOE ACTS Collection Workshop, DOE SciDAC meetings, and SC meetings.

This work was supported by the Director, Office of Advanced Scientific Computing Research (ASCR) of the U.S. Department of Energy under contract No. DE-AC02-05CH11231

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.