C++ *ABI Summary*

Revised 14 September 1999

Links

- Full open issues list.
- Full closed issues list.
- Draft data layout specification.
- Draft exception handling specification.
- Vtable layout examples.

- 64 Bit Runtime Architecture and Software Conventions for IA-64, Intel document SC - 2791, Rev. No. 2.4E.

- IA-64 Instruction Set Architecture

Meetings

When		Where	Phone
9 September	10:00-12:00 PDT	completed	
16 September	10:00-12:00 PDT	Cancelled	
23 September	10:00-12:00 PDT	SGI Sapphire 20L	650-933-7952
30 September	10:00-12:00 PDT	SGI Sapphire 20L	650-933-7952
7 October	10:00-12:00 PDT	SGI Sapphire 20L	650-933-7952
14 October	10:00-12:00 PDT	SGI Sapphire 20L	650-933-7952
21 October	10:00-12:00 PDT	SGI ???	650-933-7952
28 October	10:00-12:00 PDT	SGI ???	650-933-7952

Note: When calling the SGI telephone bridges, the first caller continues to ring until the second party joins. To get rid of it, you can call from a second phone, and hang it up once someone else calls.

Participants

Company	Name	Telephone	Fax	Email
	overall reflector			cxx-abi@corp.sgi.com
	Jim Dehnert	(650) 933-4272	(650) 932-4272	dehnert@sgi.com
	Matt Austern	(650) 933-4196	(650) 932-4196	austern@engr.sgi.com

SGI	Hans Boehm	(650) 933-7144	(650) 932-7144	boehm@engr.sgi.com
	Shin-Ming Liu	(650) 933-4287	(650) 932-4287	shin@engr.sgi.com
	John Wilkinson	(650) 933-4298	(650) 932-4298	jfw@engr.sgi.com
	reflector			cxx-abi-sgi@engr.sgi.com
CodeSourcery	Mark Mitchell	(650) 365-3064	(650) 375-7694	mark@codesourcery.com
Compaq	Ron Brender	(603) 884-2088	(603) 884-0153	brender@zko.dec.com
	Coleen Phillimore	(603) 884-0939	(603) 884-0153	coleen@philli.zko.dec.com
	Dave Plummer	(603) 884-3936	(603) 884-0153	Dave.Plummer@Compaq.com
Cygnus	Jason Merrill	(408) 542-9665	(408) 542-9765	jason@cygnus.com
	Benjamin Kosnik	(408) 542-9643	(416) 542-9765	bkoz@cygnus.com
	Ian Carmichael	(416) 482-3946	(416) 482-6299	iancarm@cygnus.com
	Ulrich Drepper	(408) 765-4699	?	drepper@cygnus.com
	reflector			c++abi@cygnus.com
EDG	Daveed Vandevoorde	(650) 592-5768	(650) 592-5781	daveed@edg.com
EPC	Colin McPhail	+44 (131) 225-6262	+44 (131) 225-6644	colin@epc.co.uk
Hewlett-Packard	Cary Coutant	(408) 447-5759	(408) 447-4244	cary@cup.hp.com
	Christophe de Dinechin	(408) 447-5491	(408) 447-4244	ddd@cup.hp.com
	Sassan Hazeghi	(408) 447-5007	(408) 447-4244	sassan@cup.hp.com
	reflector			cxx-abi-hp@cllmail.cup.hp.com
Humboldt-Universität zu Berlin	Martin von Löwis	+49 30 2093 3118	+49 30 2093 3112	loewis@informatik.hu-berlin.de
	Mark Mendell	(416) 448-3485	(416) 448-4414	mendell@ca.ibm.com

IBM	Allan H. Kielstra	(416) 448-3558	(416) 448-4414	kielstra@ca.ibm.com
	reflector			CxxABI-ADTC-CAN@ca.ibm.com
Intel	Sunil Saxena	(408) 765-5272	(408) 653-8511	Sunil.Saxena@Intel.com
	Suresh Rao	(408) 765-5416	(408) 765-5165	Suresh.K.Rao@Intel.com
	Priti Shrivastav	(408) 765-4699	(408) 765-5165	Priti.Shrivastav@Intel.com
	reflector			cxx-abi@unix-os.sc.intel.com
SCO	Jonathan Schilling	(908) 790-2364	(908) 790-2426	jls@sco.com
Sun	George Vasick	(650) 786-5123	(650) 786-9551	george.vasick@eng.sun.com
	Michael Lam	(650) 786-3492	(650) 786-9551	michael.lam@eng.sun.com
	Michael Ball	(650) 786-9109	(650) 786-9551	michael.ball@eng.sun.com
	Reza Monajjemi	(650) 786-6175	?	reza.monajjemi@eng.sun.com

Objectives

- Interoperable C++ compilation on IA-64: we want users to be able to build relocatable objects with different compilers and link them together, and if possible even to ship common DSOs. This objective implies agreement on:
  - Data representation
  - Object file representation
  - Library API

- ISO Standard C++: highest priority is functionality and performance of standard-compliant code. It should not be sacrificed for the benefit of language extensions or legacy implementations (though considering them as tie-breakers is fine).

- Some areas will be easier to agree on than others. Our priorities should be based on achieving as much interoperability as possible if we can't attain perfection. That is, it is better to end up with a few restrictions being required for interoperable code, than to have no interoperability at all. This suggests priorities as follows:
  1. Items requiring base ABI changes that might affect other languages, and will therefore become impossible soon. Examples include exception handling / stack unwind, or ELF changes (not extensions).
  2. Core features where differences will prevent virtually any C++ object code from porting. Examples include data layout and calling conventions.
  3. Limited usage features, where users can achieve portability by avoiding the feature. An example might be multi-threading.
  4. Peripheral features, where the requirements on users to achieve portability are clear and easy to implement. An example is non-explicit inlining, where compilers would presumably allow it to just be suppressed.
  5. Tool interfaces, which affect how users build code, rather than what they build. An example is the compilation command line.

- Mechanisms/methods which allow coexistence of incompatible implementations may be suitable in some cases. For instance, packaging vendor-specific compiler support runtimes in DSOs occupying distinct namespaces might allow multiple such DSOs to be loaded for mixed objects and avoid requiring that all vendors have the same

support runtimes.

Action Item Status

#	Action	Who	Status	Opened	Closed
1	Distribute Sun C++ ABI	Mike Ball	open	990603	
2	Distribute Sun C++ ABI Rationale	Mike Ball	open	990603	
3	Distribute Taligent C++ ABI	Cary Coutant	open	990603	
4	Expedite IA-64 RT Arch doc release	Cary Coutant	closed	990603	990720
5	Set up n-way NDA for eligible members	Priti Shrivastav	open	990603	
6	Organize/summarize object layout issues and alternatives	Matt Austern	closed	990603	990624
7	Write-up of Vfunc call protocol	Christophe de Dinechin	closed	990610	990805
8	Write-up of object layout strawman	Matt Austern	closed	990610	990624
9	Check with c++-core about empty base placement	Jason Merrill	closed	990610	990618
10	Describe dynamic cast / inaccessible base issue	Daveed Vandevoorde	closed	990617	990701
11	Summarize ctor/dtor issues	Michael Lam	closed	990617	990729
12	Describe Intel exception model	Priti Shrivastav	closed	990624	990818
13	Propose RTTI representation	Daveed Vandevoorde	closed	990701	990819
14	Open EDG exception stack?	Daveed Vandevoorde	open	990715	
15	Priority scheme descriptor for C-2	Jim Dehnert	closed	990715	990804
16	Covariant return scheme	Jason Merrill	closed	990715	990729
17	Validate Christophe's B-6 Vtable layout	Jim Dehnert	closed	990729	990811
18	Check Sun dynamic cast algorithm	Michael Lam	closed	990805	990812
19	Write up C-3 destructor proposal	Jim Dehnert	closed	990805	990811
20	Look at implications of discussion on B-1, B-6	Christophe de Dinechin	closed	990812	990830
21	Look at alternative implementations of C-2	Jim Dehnert	open	990812	
22	Describe HP hash processing for RTTI	Christophe de Dinechin	open	990826	
23	Complete RTTI proposal	Daveed Vandevoorde	open	990826	
24	Summarize Vtable issues	Christophe de Dinechin / Jason Merrill	open	990909	

25	Traceback personality API	Christophe de Dinechin	open	990909	

Issue Status

In the following table, the ***class*** column attempts to classify the issue on the basis of what it likely affects. The identifiers used are:

call	Function call interface, i.e. call linkage
data	Data layout
lib	Runtime library support
lif	Library interface, i.e. API
g	Potential gABI impact
ps	Potential psABI impact
source	Source code conventions (i.e. API, not ABI)
tools	May affect how program construction tools interact

#	Issue	Class	Status	Source	Opened	Closed
A	**Object Layout**					
A-1	Vptr location	data	closed	SGI	990520	990624
A-2	Virtual base classes	data	closed	SGI	990520	990624
A-3	Multiple inheritance	data	closed	SGI	990520	990701
A-4	Empty base classes	data	closed	SGI	990520	990624
A-5	Empty parameters	data	closed	SGI	990520	990701
A-6	RTTI (`type_info`) .o representation	data call ps	open	SGI	990520	
A-7	Vptr sharing with primary base class	data	closed	HP	990603	990729
A-8	(Virtual) base class alignment	data	closed	HP	990603	990624
A-9	Sorting fields as allowed by [class.mem]/12	data	closed	HP	990603	990624
A-10	Class parameters in registers	call	closed	HP	990603	990701
A-11	Pointers to member functions	data	closed	Cygnus	990603	990812
A-12	Merging secondary vtables	data	closed	Sun	990610	990805
A-13	Parameter struct field promotion	call	closed	SGI	990603	990701
A-14	Pointers to data members	data	closed	SGI	990729	990805
B	**Virtual Function Handling**					
B-1	Adjustment of "this" pointer (e.g. thunks)	data call	open	SGI	990520	
B-2	Covariant return types	call	closed	SGI	990520	990722

B-3	Allowed caching of vtable contents	call	closed	HP	990603	990805
B-4	Function descriptors in vtable	data	closed	HP	990603	990805
B-5	Where are vtables emitted?	data	open	HP	990603	
B-6	Virtual function table layout	data	open	SGI	990520	
B-7	Objects and Vtables in shared memory	data	closed	HP	990624	990805
B-8	dynamic_cast	data	open	SGI	990628	
C	**Object Construction/Destruction**					
C-1	Interaction with .init/.fini	lif ps	open	SGI	990520	
C-2	Order of ctors/dtors w.r.t. link	lif ps	open	HP	990603	
C-3	Order of ctors/dtors w.r.t. DSOs	ps	open	HP	990603	
C-4	Calling vfuncs in ctors/dtors	call	open	Cygnus	990603	
C-5	Calling virtual destructors	call	open	Sun	990603	
C-6	Extra parameters to ctors/dtors	call	open	Cygnus	990603	
C-7	Passing value parameters by reference	call	closed	All	990625	990805
C-8	Returning classes with non-trival copy constructors	call	closed	All	990625	990722
D	**Exception Handling**					
D-1	Language-specific data area format	lib ps	open	SGI	990520	
D-2	Unwind personality routines	lib ps	open	SGI	990520	
D-3	Unwind process clarification	lib ps	open	SGI	990520	
D-4	Unwind routines nested?	lib ps	open	SGI	990520	
D-5	Interaction with other languages (e.g. Java)	lib ps	open	HP	990603	
D-6	Allow resumption in other languages?	lib ps	open	HP	990603	
D-7	Interaction with signals or asynch events	lib ps	open	HP	990603	
D-8	Interaction with threads packages	lib ps	open	SGI	990603	
D-9	longjmp interaction	lib ps	open	HP	990908	
E	**Template Instantiation Model**					
E-1	When does instantiation occur?	tools	open	SGI	990520	
E-2	Separate compilation model	tools	open	SGI	990520	
E-3	Template repository	tools	open	HP	990603	
F	**Name Mangling**					
F-1	Mangling convention	call	open	SGI	990520	
F-2	Mangled name size	call g	open	SGI	990520	

F-3	Distinguish template instantiation and specialization	call g	open	SGI	990520
G	**Miscellaneous**				
G-1	Basic command line options	tools	open	HP	990603
G-2	Detection of 1-def rule violations	call	open	Sun	990603
G-3	Inlined routine linkage	call	open	Sun	990603
G-4	Dynamic init of local static objects and multithreading	call	open	SCO	990607
G-5	Varargs routine interface	call	open	HU-B	990810
H	**Runtime Library Interface**				
H-1	Runtime library DSO name	tools	open	SGI	990616
H-2	Runtime library API	lif	open	SGI	990616

Procedure Notes from 3 June 1999

- Meetings: 10-12 Thursdays at SGI for the near term.
- Intel NDA: Generally unnecessary. Priti will set up n-way for eligible members for cases where needed. Cary expects RT architecture/software conventions document to be released in the next month or two, removing most of the issues.
- Communication: Use of reflector encouraged for discussion. NDA communication will be handled with password-protected PDF once Intel sets up n-way.
- Available documents: Parties with existing, relevant documents (includes Sun, HP) will send them to group.
- Intellectual property: Participants don't expect problems with release of any of their IP. Microsoft has extensive patents in the area, but they are excessively broad (covering obvious ideas and prior art), so expectation is that they are not a problem. Nonetheless, we should be aware of them.

Please send corrections to Jim Dehnert.