C++ Resume Addendum

This is a summary of Mr. Joseph Ellsworth's Experience in the C++ programming language and associated development platforms. Additional addendums are available with information on other areas of expertise. A 10 page detailed resume is also available.

C++ is a hybrid language which is an anscester of the C programming language. Mr. Ellsworth began programming in C during 1984 and has used it or C++ ever since. Mr. Ellsworth has used both the Ansii standard and K&R style C but preferes the Ansii Standard C because of the typesafe linking it provides.

Mr. Ellsworth purchased and used the first Borland C++ compiler available for MS-DOS. He also has used several CFRONT base compilers. More recently he has used Zortech, Microsoft Visual C++ and Borland C++ professional. He has used C++ under Unix, SUN/OS, MS-DOS, Windows 3.0 & 3.1 and OS/2.

Mr. Ellsworth has used C++ on to many contracts to mention all of them here so only the most applicable ones will be mentioned.

- RPG to C++ Translation. Mr Ellsworth implemented the RPG to C++ langauge translation tool in Smalltalk. The output of this tool was standard CFRONT 3.0 compatible C++. The translated ouput programs make heavy use of a Class Library also designed and implemented by Mr. Ellsworth. The Class library designed to be Object Oriented from the ground up and used polymorphisim, dual inhertiance and overloaded operators to simplify the readability of the code using the class library. This library duplicates much of the functionallity of the AS/400 operating system. It provides features such as automatic index mainiuplation, automatic constuciton of SQL statements and automated managment of interface to the SQL engine. It also provides a number of user interface features directly tailored to closly mimic the logic paradim used by the AS/400 operating system.
- **Mr. Ellsworth worked with Northern Telecom**. as a team member he worked extensively in C++ to reverse engineer a object model for a DMS- DTOCS (DMS Table TO Object Conversion System) and then enhanced this model with substantial functional and architectural improvements. The new model was to be the basis for a new implementation of the DTOCS system.
- **Robotics / Interface & Programming** - During this project Mr. Ellsworth designed and implemented a TTL to Analog interface system. He designed and built the circuits necessary to interface standard 12 volt DC motors to a Parallel port of a IBM-PC. Using custom designed circuits he was able to accurately control these motors down to 1/40 of a revolution.

  The circuits were all controlled by a normal bi-directional PC Parallel port. The controlling software was written in C++ which used a combination of polling, timer interrupts and service interrupts to control the external circuits.
- **Graphical / Linguistic Compression & Decompression Book Displayer.** - Mr. Ellsworth designed this product to allow the distribution of text and graphics books on PC floppy instead of hard copy. This product was a descendant of the basic speed reading tutorial program. It was implemented in C++. It has the ability to read any text file and display it in a form optimized for teaching and practicing speed reading. It has the special extension of being able to process files with embedded line/ vector graphic images.

  All user interface widgets for this package were inherited from the Speed Reading Program and enhanced for operation in graphics mode.
- **Speed Reading Tutorial Program** - This product allows any text file to be used as the source material for a timed reading. It incorporates the basic features of speed reading training such as eye movement, specified line width, specified # of lines, etc. The regular use of this program can assist the user in doubling or tripling their reading speed while improving their comprehension. This product is written in C++ and is heavily object oriented. It runs on IBM-PC Compatibles. This package represents a lot of work at the widget level class library which implemented multiple pane windows, scroll bars, pop up lists, scrolling lists, etc.
- **Btreive to C++ Interface Wrapper** - This package provides a safe high level interface from C++ to the Novell Btreive engine. It is implemented in the form of a C++ class library. It incorporates features such as automatic file opening, automatic initialization of Btreive interface, automatic movement from record buffers to field buffers. It also includes a higher level class library for managing multiple record sets, etc.
- **Object Oriented Novell SQL to C++ Interface** - This package was first written in Object Pascal and later reimplemented in C++.

  This interface supplied a safe interface to the Netware XQL engine by providing automatic handling of memory buffers. XQL is a multiple record dynamic SQL server so the library had the built in ability to handle multiple record sets. Since the extract buffer format was not know until run time the XQL ability to return field names and formats of the fields in the active fetch buffer was used to dynamically link the fields in the record buffer to the internal fields via field name. There is a built in ability to automatically parse the fields out of the record buffer, do any necessary conversion and store the value in the fields buffer.

Joseph Ellsworth
5442 South 900 East #158 SLC UT 84117
801-596-9833

Thursday, April 30, 1998

C++ Resume Addendum

This is a summary of Mr. Joseph Ellsworth's Experience in the C++ programming language and associated development platforms. Additional addendums are available with information on other areas of expertise. A 10 page detailed resume is also available.

C++ is a hybrid language which is an anscester of the C programming language. Mr. Ellsworth began programming in C during 1984 and has used it or C++ ever since. Mr. Ellsworth has used both the Ansii standard and K&R style C but preferes the Ansii Standard C because of the typesafe linking it provides.

Mr. Ellsworth purchased and used the first Borland C++ compiler available for MS-DOS. He also has used several CFRONT base compilers. More recently he has used Zortech, Microsoft Visual C++ and Borland C++ professional. He has used C++ under Unix, SUN/OS, MS-DOS, Windows 3.0 & 3.1 and OS/2.

Mr. Ellsworth has used C++ on to many contracts to mention all of them here so only the most applicable ones will be mentioned.

- RPG to C++ Translation. Mr Ellsworth implemented the RPG to C++ langauge translation tool in Smalltalk. The output of this tool was standard CFRONT 3.0 compatible C++. The translated ouput programs make heavy use of a Class Library also designed and implemented by Mr. Ellsworth. The Class library designed to be Object Oriented from the ground up and used polymorphisim, dual inhertiance and overloaded operators to simplify the readability of the code using the class library. This library duplicates much of the functionallity of the AS/400 operating system. It provides features such as automatic index mainiuplation, automatic constuciton of SQL statements and automated managment of interface to the SQL engine. It also provides a number of user interface features directly tailored to closly mimic the logic paradim used by the AS/400 operating system.
- **Mr. Ellsworth worked with Northern Telecom**. as a team member he worked extensively in C++ to reverse engineer a object model for a DMS- DTOCS (DMS Table TO Object Conversion System) and then enhanced this model with substantial functional and architectural improvements. The new model was to be the basis for a new implementation of the DTOCS system.
- **Robotics / Interface & Programming** - During this project Mr. Ellsworth designed and implemented a TTL to Analog interface system. He designed and built the circuits necessary to interface standard 12 volt DC motors to a Parallel port of a IBM-PC. Using custom designed circuits he was able to accurately control these motors down to 1/40 of a revolution.

  The circuits were all controlled by a normal bi-directional PC Parallel port. The controlling software was written in C++ which used a combination of polling, timer interrupts and service interrupts to control the external circuits.
- **Graphical / Linguistic Compression & Decompression Book Displayer.** - Mr. Ellsworth designed this product to allow the distribution of text and graphics books on PC floppy instead of hard copy. This product was a descendant of the basic speed reading tutorial program. It was implemented in C++. It has the ability to read any text file and display it in a form optimized for teaching and practicing speed reading. It has the special extension of being able to process files with embedded line/ vector graphic images.

  All user interface widgets for this package were inherited from the Speed Reading Program and enhanced for operation in graphics mode.
- **Speed Reading Tutorial Program** - This product allows any text file to be used as the source material for a timed reading. It incorporates the basic features of speed reading training such as eye movement, specified line width, specified # of lines, etc. The regular use of this program can assist the user in doubling or tripling their reading speed while improving their comprehension. This product is written in C++ and is heavily object oriented. It runs on IBM-PC Compatibles. This package represents a lot of work at the widget level class library which implemented multiple pane windows, scroll bars, pop up lists, scrolling lists, etc.
- **Btreive to C++ Interface Wrapper** - This package provides a safe high level interface from C++ to the Novell Btreive engine. It is implemented in the form of a C++ class library. It incorporates features such as automatic file opening, automatic initialization of Btreive interface, automatic movement from record buffers to field buffers. It also includes a higher level class library for managing multiple record sets, etc.
- **Object Oriented Novell SQL to C++ Interface** - This package was first written in Object Pascal and later reimplemented in C++.

  This interface supplied a safe interface to the Netware XQL engine by providing automatic handling of memory buffers. XQL is a multiple record dynamic SQL server so the library had the built in ability to handle multiple record sets. Since the extract buffer format was not know until run time the XQL ability to return field names and formats of the fields in the active fetch buffer was used to dynamically link the fields in the record buffer to the internal fields via field name. There is a built in ability to automatically parse the fields out of the record buffer, do any necessary conversion and store the value in the fields buffer.

Joseph Ellsworth
5442 South 900 East #158 SLC UT 84117
801-596-9833

Thursday, April 30, 1998