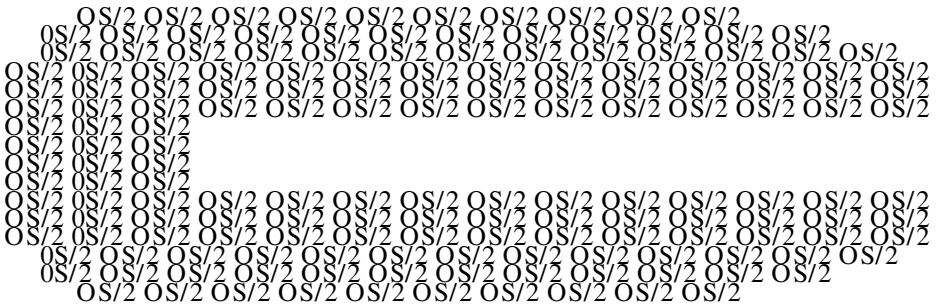

IBM OS/2 Compiler News

April 1993 No. 4

In this issue

Here we are, our first 93 issue. Thank you for all the feedback you've sent - our first year was a resounding success. It seems you like our newsletter the way it is, which is just as well because we hadn't planned to make any major changes (except for improving print quality, you'll be pleased to hear).
Please continue to write/fax to us - we welcome your comments.

- Latest beta news.....page 2
- A better C.....page 6
- First in a series of tools articles suggested by our readers, IBM OS/2[1] System Performance Monitor/2.....page 9
- Browsing C++ .....page 11
- Clarifications on a few items from our last issue.....page 28
- Q&A on C Set/2™ ... the C++ beta code ... WorkFrame/2™ .... page 29

and other assorted items and comments.

See our centre pages!

How are you getting on out there with our C Set/2? We'd like to know, and we'll reward you for telling us. See our survey and find out more ..... pages 21-24.

[1] ™ OS/2 and products marked ™ are trademarks or registered trademarks of IBM Corporation

Beta Sector

What have we been up to?

It's been five months since our C++ beta made its debut on CD-ROM, and to say it's been interesting is the understatement of the year. We thought you had plenty to say on C Set/2. Now we find you're just as vocal, if not more so, on C++. Don't get us wrong, we're not complaining...far from it. We need your feedback on our products. We take note of everything you tell us: positive or negative, it all gets used.

But first, let's clear up some confusion about CD-ROM version of the code...how many drops have there been, you keep asking, and how up-to-date are they? There have been two drops issued, and a third is in the works.

1. Drop One

   The original beta on the 11/92 PDK CD-ROM. The C++ code was dated October/November.

2. Drop Two

   A "limited edition" issued for the OS/2 Technical Interchange in Phoenix, Arizona. This was again an OS/2 PDK, and seems to be known mainly as the Phoenix PDK. This C++ code was dated December/January.

3. An upcoming drop, again on a PDK, available late March 92.

   This is where it gets slightly complicated. This PDK looks the same as the Phoenix PDK on the outside, and even has a 2/93 date on it. However, it contains the latest OS/2 and C++ code, i.e. later code than the Phoenix PDK. IBM Boca has been placing white stickers on the later PDKs to denote a new OS/2 level, and there is an asterisk by the date on the CD itself that alerts you to the fact it is one of the new PDKs. In case some slipped through the net and ended up without a sticker, you may wish to check your CD-ROM for an * to see which you have.

   As for the C++ on this latest PDK, it is March code, but it's still beta code, so don't expect to see everything you've asked about (like performance improvements, for example).

   The AT&T library code available originally with the 11/92 beta is still valid for the subsequent release. You don't need to purchase a new copy.

As before, the PDKs are available to Developer Assistance Program (DAP) members worldwide for a small charge, usually to cover shipping and handling. To find out about joining the DAP, check with your local IBM office, DAP group, IBMLINK BBS, or type GO OS2DAP on CompuServe.[2] In the USA, you can call 1-407-982-6408, in Canada you can fax your DAP inquiries to 416-946-2535. For technical support, or just to see what's going on, hang around OS2DF1 on CompuServe, or C-SET2 on OS2BBS.

So what do you think of it so far?

Putting a product out in beta is like going on a blind date. You find out all you can about the other person, take great care getting ready, cross your fingers, then head off into the unknown. You don't know if it'll be love at first sight, a pleasant outing, or a horror story. Life with C++ is like that. We've listened to all the comments about IBM and C, OS/2, support and service, pricing, and so on. We've learned everything we can about what users expect in a product, and how they feel things should work. We've taken care to offer a suite of tools, made ourselves look attractive and gone out on a CD-ROM at virtually no cost to users.

So how was our outing? Pretty much like the blind date, a good time had by most people, with some falling madly in love, and some saying ho-hum. Overall, we're exceedingly pleased with the results, especially as the horror story doesn't seem to have materialised.

Here are a few of the main areas of comment:

- **Documentation lacking/needs improvement**

  The code was in beta, and the manuals were too. In the first beta, there was no documentation for the PM Class Library, and we had several different print formats.

  That was then; this is now. We've made a lot of improvements, and you'll notice quite a difference in the material on our later beta. When the product is available, the documentation will be available in various formats in softcopy, hardcopy, and CD-ROM.

[2] ™ Trademark or registered trademark of CompuServe Inc

- **ICLUI is very complex**

  The lack of a User's Guide contributed to the complexity of learning this PM class library, but even with documentation, this is a complicated area to work in, and takes time to learn.

  The complexity is a reflection of the robustness of the class library. To minimize the impact to new users of ICLUI, we've created a User Guide that has 6 functional sample applications, from barebones to a sample application that demonstrates a large portion of the classes. This UG is available in the latest beta.

  Also, based on beta feedback, we've restructured the Reference Manual. Amongst other things, information is now easier to find.

  Many users expressed the need for a one page view of the hierarchy. In response, we're preparing a poster that shows you this. Watch for further info.

- **EXE size**

  Too large in many places.

  Yes, we know, but we can tell you good news. Since the first beta, there have been major improvements in exe size with debug information turned on. You may consider exe size a little larger than other vendors' products, but IPMD does offer in return rich function.

  Exe size with the use of ICLUI will be alleviated by the use of DLLs.

- **Performance**

  It will improve, won't it?

  We always try to get things right first, then make them fast. We already produce the fastest executing code, and compile time is an area we've been working hard on since the last beta, and it will decrease in the General Availability (GA) product.

The OS/2 Technical Interchange (OTI) in Phoenix

Have compiler, will travel seems an apt description of the C++ team. In February/March, we were at the OS/2 Technical Interchange in Phoenix, giving presentations and running labs, as well as manning a display area. The extended team included folks from

Lexington and Cary, so we were able to demo all parts of the product.

We opened the doors at 1 pm on Day One, for an overview presentation, followed by demo and lab sessions on migration and performance. A veritable horde crowded in and we were soon at capacity. This was the case with all the sessions, and if you couldn't get in, we apologise. At the next OTIs, we'll have bigger rooms (see the end of this article for details on future OTIs).

Despite mechanical problems - a green only screen, a shortage of hands-on machines for the Lab, and a short microphone cable that nearly throttled the speaker - the audience enjoyed what they saw, and the Q&A sessions were particularly busy. It proved to be a very good idea to run sessions on ICLUI and our other class libraries...we couldn't tell the audiences enough about using our ICLUI, and you couldn't thank us enough for providing the only PM encapsulation library on the market today. You're welcome!

The display area attracted a lot of visitors[3] some of whom were seeing C Set+ + for the first time. On the other hand, it was quite clear from the very detailed questioning (and occasional downright interrogation) that many of you have been through C Set/2 and C Set+ + with the proverbial fine tooth-comb. We showed all the components of the product, in particular IPMD, the debugger, as we had two of the Lexington programmers at the show. And with several of the developers of ICLUI in from Cary, plus the people from Toronto, we were able to handle most of the things you asked. If you left us a query, either verbally or on a card or such, we're working our way through them. Bear with us.

Overall, the show was a success for C Set+ + , and it was a pleasure for us to meet in person many of the users we've been corresponding with for so many months on the various electronic networks.

If you didn't get to Phoenix, don't worry. More OTIs are planned, and the next two are Toronto, Canada (June 21-25) and Orlando, Florida (August 29-Sept 2). There are European and Asia/Pacific OTIs planned. We'll give you the dates when we know them. See you there!

---

[3] Our PRGS Toronto Lab poster of the "Blue Ninja" may have been a contributory factor in the number of visitors who stopped by! The poster is a limited edition, especially produced by this Lab as a gesture of appreciation for those people who fill in the survey in this newsletter. If you saw it and liked it, you know what to do....

Phoenix Q&A

Questions asked of us in Phoenix covered a wide variety of subjects. Here are a few of the most asked ones:

- **What's the name of the product?**

  C Set++ ™ is the name of the complete package, i.e. the compiler, debugger, browser, trace analyzer, and class libraries.

- **When is C Set++ going to be available?**

  We expect it to be available by the end of 1H93.

- **How much will it cost?**

  The product will be very competitively priced. Comparing what functions/features there are for the price, it is good value indeed.

- **Will there be upgrades from C Set/2?**

  You bet. We plan to offer competitive upgrades.

- **Why are the betas only on CD-ROM?**

  As the PDKs are being issued on CD-ROM, we took the opportunity to go along with our Boca colleagues and make our beta available quickly and easily on this popular medium.

- **Will you make ICLUI available on other platforms?**

  We're very actively considering this as we realise there is considerable customer interest in other platforms, such as AIX™.

- **Will you make a 16 bit C Set?**

  No, we don't intend to. Although we accept the existence and long life of 16 bit, we prefer to look forward in our choice of platforms.

- **What about other class libaries?**

  What about them? IBM is interested in having other class library vendors port their products to our compiler. We welcome inquiries.

From C to Shining C

While C++ support is the centrepiece of our new compiler product,we haven't forgotten our existing C Set/2 customers - we've included plenty of new goodies on the C side of our compiler. Some of them are also available in C++, but some are specific to C. Here's a brief summary:

New Language Features

Several new features increase our compatability with other popular compilers:

- all C Set/2 migration mode features (except unsupported keyword removal) moved into extended mode (selected by /Se)
- anonymous unions
- enumeration size is controllable through /Su option, and may now by 1, 2, or 4 bytes, or the smallest size necessary to contain the enumeration constant range
- #pragma import - specify function is to be imported from a DLL
- _Export/#pragma export - specify function is to be exported from a DLL

Virtual Device Driver (VDD) Support

A major shortcoming of the original C Set/2 product was the inability to produce 32-bit VDDs. That's been remedied with a complete package of compiler support in C Set++, and C Set++ capable VDD headers in the OS/2 Developer's Toolkit included with C Set++. The compiler support includes:

- _Pascal and _Far32 _Pascal linkages
- /Gv - save, reload, and restore ES and DS segment registers on all entry points - this is our equivalent to _loadds
- /ND - support for renaming the data and constant segments
- /NT - support for renaming the code segment
- #pragma entry - specify which function the system should pass control to
- low level Built-In Functions (BIF) - these include _interrupt, _enable, _disable, _inp, _outp, and several others, including one to access the TIB

New Debugging and Listing Features

As many of our customers requested (ok, demanded is more like it), we've divided the extended warning messages previously controlled by the /K options and checkout pragma into smaller groups

- /Lx - xref listing containing only referenced variables
- /La - aggregate map containg only referenced variables
- #pragma info, /Wgrp - control over groups of extended warning messages
- /Pe - produce preprocessed output without #line directives

C Library Improvements

You're not only getting new functions, you'll notice tighter EXE size, made possible by recoding some functions, and by making some little-used support optional

New functions

- _msize
- _threadstore - access thread-specific storage

Optional Support

- /Sv - include memory file support
- /Tx - include full exception dump capability

Optimizations

We're constantly trying to generate the best 32-bit code on OS/2 2.x platforms. We've added several new optimizations for this release and enhanced some of our existing support:

- Pentium™ [4] support - you'll be ready when your hardware arrives
- instruction scheduling for both the Pentium and the 486
- user function inlining - automatic inlining and selected functions only
- intermediate code linking - by merging the intermediate code for your complete application before code generation, better automatic inlining, register allocation, alias analysis, and data packing can be achieved. As a bonus, the intermediate code linker also will diagnose many inter-file programming errors, and discard unused code and data.
- floating-point transcendental Built-in Functions
- /Gi - fast integer mode yielding faster shifts
- better floating-point register allocation
- improved loop unrolling

Many of these enhancements were suggested by our users. We'd like to thank you for your suggestions, and urge you to keep them coming. An excellent way to contact us is through the comment form at the end of this newsletter.

[4] Trademark of Intel Corp

IBM System Performance Monitor/2 Version 2.0 (SPM/2)

Here we are, providing you with the wherewithal to code OS/2 2.0 applications, but once you're done, what about performance? And not just for standalone..what about your LAN environment? Have we got a product for you: SPM/2.

Announced in October 1992, SPM/2 provides an integrated set of performance monitoring and analysis facilities for the OS/2 2.0 environment. Its capabilities?

- Collects OS/2 2.0 CPU, Memory, File, Swap file, Cache, Disk, Printer and Communications Port utilization data
- Remote collection from OS/2 2.0 LAN Server and Requester workstations
- Graphing Facility displays realtime CPU, Disk, RAM activity, or plays back previously recorded data.
- Workstation, application, process and thread-level reporting from recorded data
- Memory Analyzer provides information via a Presentation Manager and Hyperblock interface, and includes Tutorial/Helps on OS/2 Memory Mgt
- Support for user-defined performance metrics ("user hooks")

Why should you use SPM/2? For these reasons:

- With the kind of performance data SPM/2 gives you, Administrators and Performance Analysts can isolate problems more quickly and accurately, resulting in a quicker resolution. By accurately narrowing in on the cause of a performance problem, SPM/2 can also help you avoid investing in costly hardware solutions that do not solve the performance problem.
- Application Developers can potentially produce more efficient applications, more quickly, by using SPM/2's performance information and application-specific user hooks to narrow in on defects and choose smarter application designs.

SPM/2 is order number 5871-AAA. Please go through your regular IBM channels for more information on this product.

DeScribing C Set/2

On March 1st, DeScribe Inc. announced that it is now shipping its new DeScribe[5] Word processor 4.0 Subscription Edition (SE) Revision C, the only word processor, say DeScribe, written specifically for OS/2 2.0. DeScribe Word Processor combines advanced word processing functions with powerful desktop publishing features and extensive drawing capabilities. Why does this appear in OS/2 Compiler News? Because this latest revision exploits the C Set/2 compiler and optimizer ..... ***'DeScribe has just started shipping Revision C of our new DeScribe Word processor Version 4.0. This revision uses IBM's new C Set/2 32-bit compiler and now takes advantage of the compiler's optimization capabilities. We have experienced as high as 40% increases in performance of some of the more compute intensive operations in DeScribe. Because it is important for our customers to get the benefits of this new revision, we are sending free upgrades to all registered users of DeScribe 4.0 and all users that received 4.0 upgrades." - Allen Katzen, President and CEO, DeScribe Inc.***

Readers interested in knowing more can contact DeScribe at 916-646-1111 (voice) or 916-923-3447 (fax).

Ouch!

Print quality

Some readers have contacted us about the poor quality of their last copy of the newsletter, number 3 dated November 92. We can't do much about copies of copies that float around except to say that if you contact us, we'll ship you a good copy. We appreciate your good humour and patience when you receive one of these copies: I guess being copied unto the nth generation is a feather in our cap.

As for the quality leaving here, we did have some fade problems in one of our runs, and some inside pages were an exercise in eye testing. We apologise for that, and trust you find this copy much more readable - we changed our process to improve print quality.

[5] ™ DeScribe is a trademark of DeScribe Inc of Sacramento, CA.

You want it when??

The recent mail also has a lot of queries as to how a November 92 issue turned up in people's mailboxes in March 93. Quite simple, really, you will be pleased to know. We received a dump of the AT&T library requests in 2/93 from our US 1-800 number and added the data to our mailing list. To welcome you to the C Set family, we shipped out our latest newsletter, which happened to be November 92. (The queries were asking for more and later newsletters, so we don't feel so bad.)

Lastly, on distribution, readers are welcome to ask for back copies. Just write and ask.

Browsing C++

At various shows and conferences, we have been showing a piece of technology that was not in the beta: a browser. It has drawn a lot of comment and as a result, we offer now a review of our Browser technology on OS/2.

Abstract

The IBM OS/2 C++ Browser is a sophisticated, post-compilation, static-analysis tool which complements the compiler by supporting full-function browsing of C++ programs and class libraries. It enhances program comprehension by providing alternative ways of viewing the semantic relations of program components (e.g., class hierarchies, call graphs, cross references, and source-text definitions).

The Browser is written in C++, and features a multi-threaded, graphical user interface (GUI). The GUI has been carefully designed for both the novice and expert. Its design strikes a balance between ease-of-use required by the novice user, and flexibility and power demanded by the expert user.

Background

The ability to effectively browse source code originates from a programmer's need to understand program structure at different conceptual levels. Many vendors, recognizing this need, tend to provide extensions to the ubiquitous textual editor. Unfortunately, this approach does not address the fundamental requirement to view semantic relations in the manner most suited to the particular relation. For example, a textual editor, though good at providing a view of function or class implementations, is not good at providing a view of a class hierarchy or call graph. Moreover, it is not good at providing an abbreviated view of the source text as a list of compo-

nents (e.g., the names of all member functions of a particular class, or all globally defined type names). We believe a good browser must address the issue of providing alternative ways of viewing semantic relations.

There are two main components to the IBM OS/2 C++ Browser:

- a back-end database
- a front-end GUI.

The OS/2 C++ Browser back-end is a port of the AIX C++ Browser back-end [1, 2]; the OS/2 C++ Browser front-end is a redesign of the AIX C++ Browser front-end. The OS/2 C++ Browser exploits OS/2 features, such as multi-threading and Presentation Manager (PM), and conforms to OS/2 standards, such as Common User Access (CUA).

Description

A browser must be intuitive enough that a novice will want to use it, but powerful enough that an expert will continue to use it. This goal was achieved was by carefully considering usability issues during the design of the OS/2 C++ Browser GUI. The sections that follow will outline the features provided by the IBM OS/2 C++ Browser.

Multiple Windows

The Browser is a multi-window application with four types of window: Control, Graph, List, and Text. A single Control window is provided to manage all Graph, List, and Text windows and to centralize access to global database operations. Any combination of Graph, List, and Text windows are provided to support viewing of the various semantic relations defined on a program's components. Each type of window will be described in the following sections.

Control Window: The Control window (see Figure 1 on page 13) is the initial window that appears when a Browser session begins. Global tasks, such as browser file loading and unloading, and opening new Browser windows, are performed in this window. The client area of the Control window displays an icon for each Browser window currently available. The icon display can be laid out in one of several standard OS/2 styles, and the individual icons can be used to manipulate their associated window (e.g., restore, minimize, maximize, or close the associated window). In effect, the Control window acts as a Browser-specific desktop.

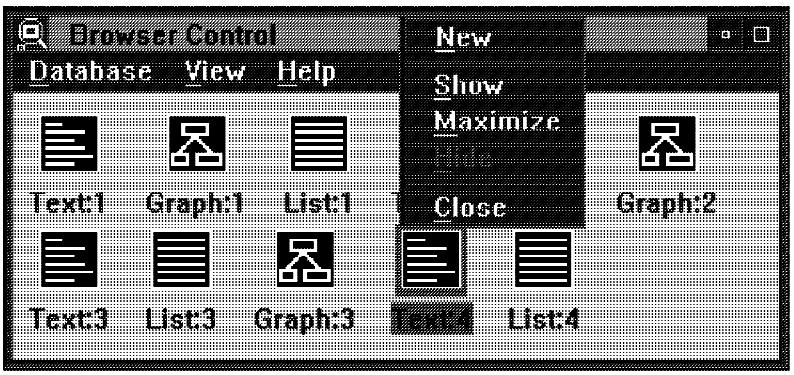

Figure 1. Control Window

Graph Window: A Graph window (see Figure 2) provides a graphical representation of a semantic relation. It is particularly well-suited for representing relations between program components, for example, the relation between a base class and its derived classes, the relation between a function and its callers, the relation between a file and those it includes, and the relation between a program component and those it contains.

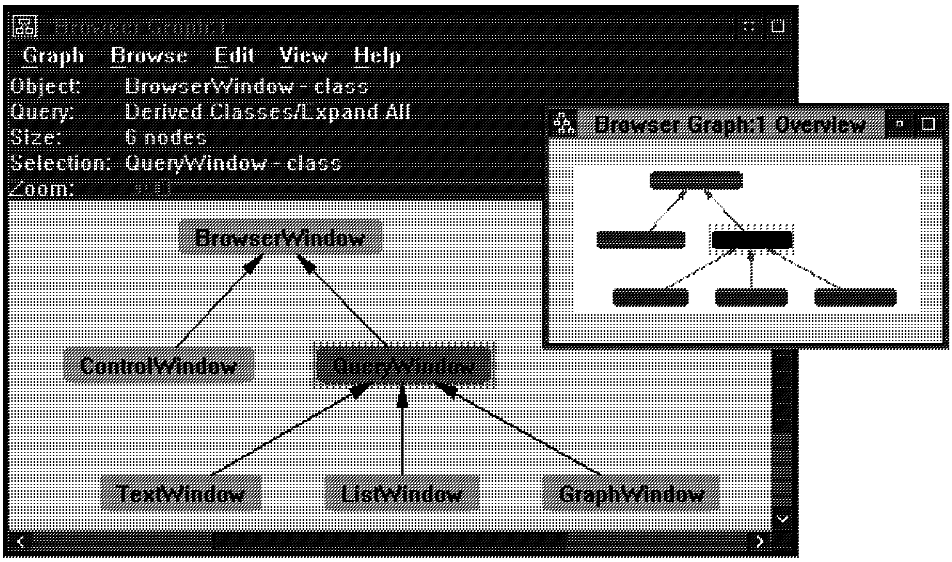

Figure 2. Graph and its Overview Window

To support viewing of large graphs, Graph windows provide an Overview window (see Figure 2). The Overview window displays a miniature view of the entire graph and highlights the portion currently visible within the Graph window. The visible portion of the graph can be modified by resizing the highlighted portion in the Overview window, by adjusting the zoom slider, by incrementally

adjusting the zoom level using menu actions, by selecting an area in the Graph window using the mouse, or by adjusting the scroll bars. In addition, portions of the graph can be elided to reduce the overall size of the graph, and various layouts styles and algorithms are supported [3].

List Window: A List window (see Figure 3) provides an abbreviated view of a set of program components (e.g., all classes, all members of some class, all function templates, and all global variables), and can optionally display the kind (type) of each listed object. For example, the list window can be used to display the members of a particular class that are available to itself (e.g., private, protected and public), its derived classes (e.g., protected and public), or any client of the class (e.g., public). It is important to note that this type of information is not readily available just by perusing source text; the shear volume of source text can be overwhelming whereas a List window presents the desired information at a glance.

Figure 3. List Window

Text Window: A Text window provides a way of viewing a program's source text. For a given object, it can display the source text of that object's definition, declaration, or usage, thereby clarifying questions about scope. For example, the programmer can determine which definition or declaration an overloaded function call uses. Text windows also allow the programmer to modify source code. Moreover, it can be used to determine the impact on compilation of modifying a class definition.

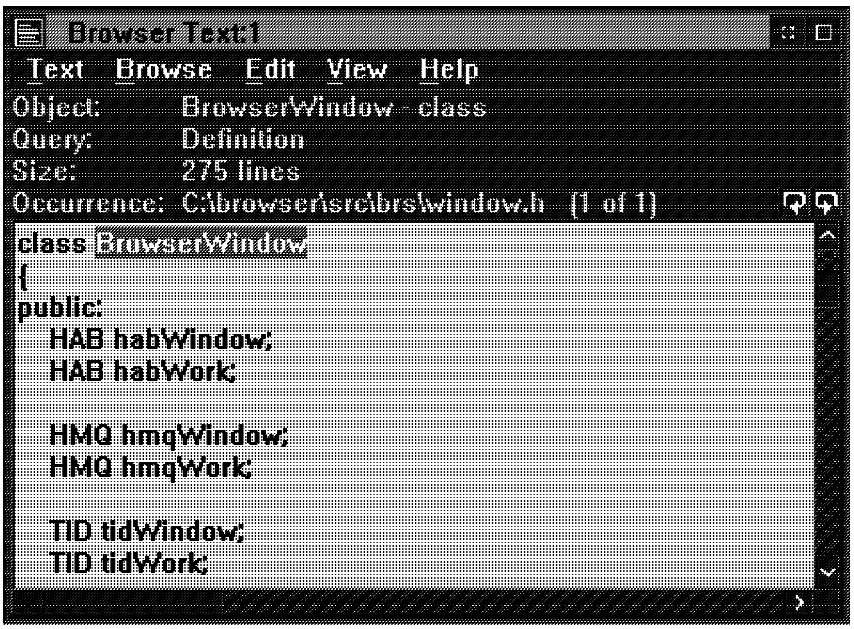

Figure 4. Text Window

Searches and Queries

The Browser yields information about source programs using either a database search or an object query. Each operation is applicable in any Graph, List, or Text window. A database search browses all facts in the database and is therefore the starting point for viewing program components. An object query typically follows a database search allowing the programmer to browse only those facts in the database that relate to the particular object in question.

For example, a database search could be used to obtain a list of all classes, and a subsequent object query could be used to obtain a list of all members of a particular class.

Each type of window has its own particular searches and queries. The type of query available also depends on the object being queried. A global clipboard, called the Object List, is shared by all the windows and allows an object displayed in one window to be queried in another. A single uniform interface mechanism, which is available in each Graph, List, and Text window, is provided for formulating searches and queries. Its design is described later in this article.

Search and Query Filters: A database search is formulated by specifying three levels of filter; an object query is formulated by specifying an object and three levels of filter. The three filter levels are called

Scope, Category and Attribute. Each filter allows the programmer to restrict the amount of information retrieved from the database.

In a Graph window search, the filters could be used as follows: Scope to specify an inheritance graph, Category to specify only root nodes, and Attribute to specify only classes with base or derived classes. In a List window query applied to a class, the filters could be used as follows: Scope to specify a members list, Category to specify only function members, and Attribute to specify only static members.

Search and Query Dialog: The Browser possesses a novel dialog interface common to all Graph, List, and Text windows which is used formulate searches and queries. This dialog solves the problem posed by deeply-nested cascaded menus with numerous choices at each level. It strikes a balance between the ease-of-use needed by the novice and the speed and flexibility demanded by the expert.

The Search dialog (see Figure 5) contains three colour-coded pushbuttons Scope, Category, and Attribute for each filter, a current selection indicator adjacent to each pushbutton, a listbox of available selections, and a group of action buttons. In addition, the Query dialog (see Figure 6 on page 17) contains a drop-down listbox which displays the current query object and provides access to other objects in the Object List.

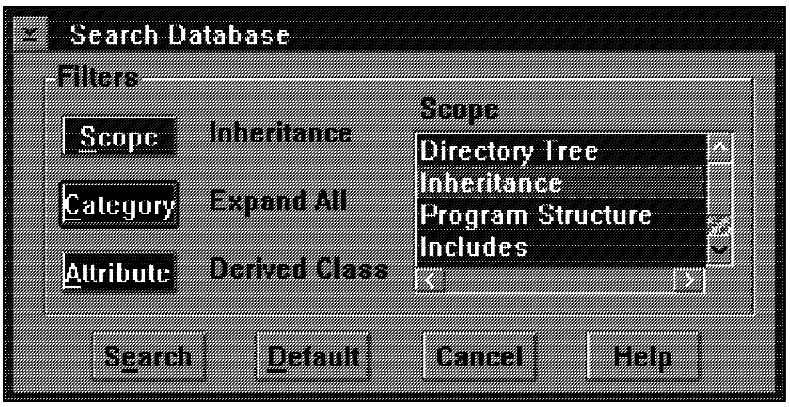

Figure 5. Search Dialog

When the dialog is initially displayed, the listbox of available choices is filled with the appropriate Scope choices, its background is coloured to be the same colour as the Scope button, and its title is set to be Scope. Also, a default choice is displayed for each Scope, Category, and Attribute button as indicated by their adjacent text. Push-

buttons not used for a particular search or query are greyed out (e.g., when no applicable filter choice is available).

Figure 6. Query Dialog

When a Scope choice is highlighted in the listbox of available choices, the text field adjacent to the Scope button is updated with that selection, and the text fields adjacent to the Category and Attribute buttons are updated to a suitable default. After a Scope choice is selected, the listbox of available choices is refilled with Category choices, it is recoloured to the colour of the Category button, and its title is changed to Category. Category and Attribute choices work in an analogous manner to Scope.

The user can, at any point, go back to a previous level of filter by clicking the appropriate button (e.g., clicking the Scope button refills the listbox with Scope choices). When the formulation of the search or query is complete, it can be issued by clicking the appropriate action button.

Object List: As mentioned previously, the Browser has a shared clipboard of program components called the Object List. The Object List allows program components displayed in one window to be copied to this clipboard so that they can subsequently be queried in another window. For example, a particular class displayed in a Graph Window can be copied to the Object List and then used in a List window to obtain a list of its members, or in a Text window to obtain its source declaration.

The Object List is available using the drop-down listbox in any Query dialog (see Figure 7 on page 18). On invoking the Query

dialog, a local copy of the global Object List is made to avoid the problem of modifications to the global Object List by other windows.

Figure 7. Object List displayed in Query Dialog

Multiple Threads

The Browser is a multi-threaded application which uses PM message queues to communicate between threads. The design of an appropriate thread structure is essential for every PM application; there should always be a thread responsible for monitoring user input, and it should remain responsive to subsequent input. Hence, any user action that might take a significant amount of time to process should be processed in a separate thread.

The Browser provides one Main thread, and a pair of threads for each window. The Main thread coordinates the interaction between the various window threads, and provides operations that should never occur simultaneously in two separate threads. For example, if some window would like to perform an operation that requires mutual exclusivity, such as adding an object to the global Object List, it simply requests that operation by posting a message to the Main thread's queue.

The pair of threads associated with each window, called the Window thread and the Work thread, also have specialized responsibilities: the Window thread monitors user requests, performs simple operations, and forwards complex operations to the Work thread; the Work thread performs asynchronous operations of arbitrary complexity (e.g., database access).

Class Hierarchy

The Browser is implemented using C++ and therefore implements its windows using a class hierarchy. Information common to all windows is stored in class BrowserWindow, including member variables for anchor blocks, message queues, thread identification numbers, and various window handles. Furthermore, function members for operations that are supported for all windows are also defined in class BrowserWindow.

Class ControlWindow is derived from BrowserWindow. It contains additional member variables and functions to manage the icon display, and to support the loading and unloading of browse files. Also derived from BrowserWindow is class QueryWindow. It contains information common to all windows displaying database information, including window handles of common status areas, and search and query settings.

Class GraphWindow, ListWindow, and TextWindow are derived from QueryWindow, and contain information specific to each particular database view. For example, GraphWindow includes information indicating the current selected node, the current zoom level, and the handle of the overview window; ListWindow includes information indicating the sort order and qualification level of the list; and TextWindow includes information indicating the update state of the displayed text and the number of occurrences of a particular object (see Figure 2 on page 13).

Conclusion

Usability tests support the claim the IBM OS/2 C++ Browser is an effective tool for browsing C++ programs. It combines ease-of-use with power and flexibility. We have even used the Browser to browse the Browser's source code.

Our work on browser technology continues with an emphasis on improving performance and providing additional power in the form of new searches and queries. We also hope to port the technology across multiple platforms to provide better tools to the ever increasing community of programmers who embrace C++ as the programming language of choice.

References

[1] K. Yasuda, BRS to Prolog Translator, Tokyo Research Laboratory, IBM Japan.

[2] H. Nakamura, Embeddable Prolog Interpreter for Database-Intensive Applications, Tokyo Research Laboratory, IBM Japan.

[3] Vance Waddle, Nodes and ARCs Graph Toolkit API Defintion, Yorktown Heights.

And now for a public service announcement...

If you are compiling large source files, or making use of malloc/calloc/realloc, you need to know about a possible problem in the C Set/2 product and the February 93 PDK (non-Phoenix version) of C Set+ + . This problem may have several different symptoms:

1. the compiler may trap during compilation. Remember, this is only for compilations of large source files
2. user's code may trap in the _heapmin_int library routine called from malloc/calloc/realloc
3. if a free or a realloc is performed, storage previously obtained through malloc/calloc/realloc may be returned to OS/2 or allocated in a subsequent call to these routines. This error is likely to occur only after large, complex combinations of calls to the C memory management routines

A fix is available for the problem in C Set/2, the product you can buy today. The fix is in CSD44. Please install.

For C Set+ + , the problem will be solved in the GA version, but in the meantime, there is a workaround (and soon a fix) for symptoms 2 and 3. The workaround applies only to code that is statically linked to the C run time library. The fix is valid for both static and dynamic linking. If you need immediate C Set+ + relief, use the workaround until the fix is available, but be aware that the workaround may greatly affect the amount of memory your application uses. Please remember that you cannot install CSD's to the beta code, so don't try it!

As soon as the fix is available, we will notify users via CompuServe, the OS2BBS C-Set2 forum, and the EMEA DAP network.

Workaround: This workaround only applies if you statically link the C-SET runtime libraries. Create heapmin.c with the contents shown below.

heapmin.c
```
---------
 #include < stdlib.h>

 int _debug_heapmin(const char *p,size_t q)            { return(0);}
 int _debug_heapmin_int(const char *p,size_t q,int x) { return(0);}
 int _heapmin(void)                                    { return(0);}
 int _heapmin_int(int x)                               { return(0);}
```

Compile this source as: icc /c /Gn+ heapmin.c

Specify heapmin.obj with your other objects during the link step, and add /NOE to your linker options.

Here's your chance ...
Tell us what you think !

You can't say that the C Set team isn't interested in what our users think. We've always been keen to hear from you about our code, the function we offer, and your ideas for the future. Feedback comes to us in many ways; this time we're asking you to take the time to fill in a survey for us.

What do we want to know this time ? Basically, who's using our product for what purposes. We added C+ + and other functions to C Set/2 because that's what you said you wanted. We'd like to make sure that, with C Set+ + , we head off in the right direction. To do that, we need to know more about who you are and what you're doing with our product. The information you give us will be treated in confidence.

As a thank you for your involvement in the survey, you will receive a copy of the

Toronto PRGS Lab "Blue Ninja" poster

(Please allow 6-8 weeks for delivery)

Please fax your completed surveys to the workstation Languages Planning department in Toronto. Alternatively, you may mail to the address shown below.

Fold, seal and mail
Or return by Fax(*): Canada 416-448-6057

(*)Please mark your fax for the attention of Ed Lynch.

- - - - - - - - - - - - - - - - - - -(Fold along this line Fold along thi

Name: _____

Address: _____

_____

Country: _____

IBM Canada Ltd.
Workstation Languages Planning
Dept. 394
22/394/844/TOR
844 Don Mills Road
North York
Ontario, M3C 1V7
Canada

Name _____ **Title** _____

Company _____ **Dept.** _____

Address _____

_____

Country _____ **PostCode** _____

Telephone _____ **Fax** _____

1. Is this your home or business address ?

□ Home □ Business □ Both

2. How many employees are in all locations of your company ?

□ 1 - 10 □ 11 - 50 □ 51 - 100 □ 101 - 500 □ > 500

3. Are you using C Set++ at home, at work, or both ?

□ Home □ Business □ Both

4. How many people in your development team ?

□ 1 - 5 □ 6 - 10 □ 11 - 50 □ > 50

5. How would you describe the type of business you're in ?

□ Corporate (non-software-related)
□ Education
□ Student
□ Government
□ Independent Software Vendor
□ Other _____
□ Consulting Services
□ Prof. practice (medical, law, etc.)

6. Which of the following best describes your title/role ?

□ Programmer
□ Technician
□ Manager
□ Other _____
□ Analyst
□ Consultant
□ Director
□ Engineer
□ Supervisor
□ Student

7. How would you characterize the type of software you (or your team) produces

□ Database
□ System software
□ Games & Entertainment
□ Other _____
□ Multimedia
□ Communications
□ Business Applications
□ Utilities
□ Development Tools
□ Information Management

8. Do you (or does your development team) create applications for:

□ Personal use at home or at work
□ Other _____
□ Internal use by others in your company
□ Software for sale by your company

9. What is your primary development platform (Select only one response) ?

□ OS/2 V2.x
□ Windows 3.x/DOS
□ Windows NT
□ VM
□ MVS
□ AS/400
□ AIX
□ Unix
□ Other _____

10. What is(are) your target execution platform(s) (You may check more than one box) ?

□ OS/2 V2.x
□ Windows 3.x/DOS
□ Windows NT
□ VM
□ MVS
□ AS/400
□ AIX
□ Unix
□ Other _____

11. When doing software development, do most of your tools reside on:

□ Desktop
□ Some tools on desktop, some on local server
□ Local server or host
□ Some on desktop, some on remote server
□ Remote server or host
□ Other _____

12. Consider the most recent application you (or your development team) created. Was it coded in (You may check more than one box):

□ C
□ C++
□ COBOL
□ RPG
□ PL/I
□ Fortran
□ Pascal
□ Assembler
□ Other _____

13. Do you (or does your development team) connect to any public bulletin board services (BBS) ?

□ We are not connected to any public BBS
□ Fidonet
□ CompuServe
□ IBMLink CFORUMs
□ Internet
□ Other _____

14. From which of the following do you currently purchase your application development tools:

□ Through company central purchasing
□ Software catalog (e.g. Prog. Paradise, etc.)
□ Vendor direct (e.g. 1-800, sales reps, etc.)
□ Value-added resellers
□ Retail store
□ Other _____

15. From which of the following would you prefer to purchase application development tools:

- ☐ Through company central purchasing
- ☐ Software catalog (e.g. Prog. Paradise, etc.)
- ☐ Vendor direct (e.g. 1-800, sales reps, etc.)
- ☐ Value-added resellers
- ☐ Retail store
- ☐ Other _____

16. In which of the following periodicals would you expect to learn about new development tools (You may check more than one box) ?

- ☐ Byte
- ☐ Microsoft Journal
- ☐ .EXE
- ☐ PC World
- ☐ Dr. Dobbs Journal
- ☐ Journal of Object-Oriented Programming
- ☐ Datamation
- ☐ PC Magazine
- ☐ Computer Language
- ☐ Information Week
- ☐ PC Week
- ☐ Other _____

17. On what medium would you prefer to receive software products and updates ?

- ☐ Standard diskettes
- ☐ Tape
- ☐ CD-ROM
- ☐ Magneto-optical disks
- ☐ Electronically
- ☐ Other _____

18. Does your group use formal project scheduling for all major application projects ?

- ☐ Currently we do and will continue to do so
- ☐ Currently we do, but won't continue to do so
- ☐ Do not currently, but will in two years
- ☐ Do not currently, and will not in two years

19. Does your group use a formal application development methodology for all major projects ?

- ☐ Currently we do and will continue to do so
- ☐ Currently we do, but won't continue to do so
- ☐ Do not currently, but will in two years
- ☐ Do not currently, and will not in two years

20. Do you use a formal code and requirements change process for all major projects ?

- ☐ Currently we do and will continue to do so
- ☐ Currently we do, but won't continue to do so
- ☐ Do not currently, but will in two years
- ☐ Do not currently, and will not in two years

21. Thinking about the applications that your group is responsible for, would you say the MAJORITY execute on: (Select only one response)

- ☐ STAND-ALONE PCs or Workstations
- ☐ One or more MAINFRAME computers
- ☐ NETWORKED PCs or Workstations
- ☐ Mix of MINIs & MAINFRAMEs
- ☐ One or more MINI computers

22. IN TWO YEARS, would you say the MAJORITY of your group's applications will execute on: (Select only one response)

- ☐ STAND-ALONE PCs or Workstations
- ☐ One or more MAINFRAME computers
- ☐ NETWORKED PCs or Workstations
- ☐ Mix of MINIs & MAINFRAMEs
- ☐ One or more MINI computers

23. What type of network server do you use ? (If more than one, what type is "primary" or "largest" ?) (Select only one response)

- ☐ PC or Workstation server
- ☐ PEER to PEER with no server
- ☐ MINI computer server
- ☐ No network
- ☐ MAINFRAME computer server

24. What type of network server will you use IN TWO YEARS ? (Select only one response)

- ☐ PC or Workstation server
- ☐ PEER to PEER with no server
- ☐ MINI computer server
- ☐ No network
- ☐ MAINFRAME computer server

25. In which of the following languages would you say the MAJORITY of your group's applications will be coded IN TWO YEARS ?

- ☐ C
- ☐ C++
- ☐ COBOL
- ☐ RPG
- ☐ PL/I
- ☐ Fortran
- ☐ Pascal
- ☐ Assembler
- ☐ Other _____

26. For each of the following, please indicate what you expect your group will do, over the next two years, with respect to the applications your group is responsible for ?

- **Replace your MAINFRAMEs or MINIs with smaller systems ? ☐ Y ☐ N**
- **Put most NEW application development on smaller systems ? ☐ Y ☐ N**
- **Move some EXISTING applications to smaller systems ? ☐ Y ☐ N**
- **Move PARTS of EXISTING applications to smaller systems ? ☐ Y ☐ N**

27. Please indicate *how important* each of the following sources of application development tool information are to you (1= Extremely;4= Unimportant).

• User group meetings	1	2	3	4
• User group publications	1	2	3	4
• Vendor conferences	1	2	3	4
• Computer industry trade publications	1	2	3	4
• Computer industry trade shows	1	2	3	4
• Consultants and industry experts	1	2	3	4
• Advertising	1	2	3	4
• Vendor direct sales reps	1	2	3	4
• Other users in your organization	1	2	3	4

28. How important is each of the following development tools and services to you: (1= Extremely;4= Unimportant)

• Object-oriented analysis/design	1	2	3	4
• Visual GUI builder	1	2	3	4
• Language-sensitive editor	1	2	3	4
• C++ interpreter	1	2	3	4
• C++ compiler	1	2	3	4
• Source-level debug	1	2	3	4
• C++ class browser	1	2	3	4
• Portable C++ class libraries	1	2	3	4
• Profiler	1	2	3	4
• Assembler	1	2	3	4
• Tool integrator (e.g. workbench)	1	2	3	4
• Automated test case generation	1	2	3	4
• Source code management/control	1	2	3	4

29. What is your level of expertise with: (1= Expert;4= Novice)

• Object-Oriented Analysis	1	2	3	4
• Object-Oriented Design	1	2	3	4
• Object-Oriented Programming	1	2	3	4
• C++	1	2	3	4

30. Have you used C Set/2 and/or WorkFrame/2 ? □ **Yes** □ **No**

If you answered "No" to the previous question, please stop here.

---

31. What attracted you to C Set/2 and/or WorkFrame/2 ?

□ Advertising
□ Developer Assistance Program (DAP)
□ Peer recommendation
□ IBM's reputation
□ References on Internet or other BBS
□ Other _____

32. Have you integrated any non-IBM tools into the WorkFrame ? If so, what sort of tool ?

□ Editor
□ Configuration management tool
□ Compiler
□ Other _____
□ Debugger

33. Please indicate *how important* each of the following is to you, and *how satisfied* you are with the C Set/2 and WorkFrame/2 solution IBM has provided.

(1= Extremely Important; 4= Unimportant) (VS= V. Satisfied; VD= V. Dissatisfied)

Tools allow developer to quickly modify applications	1	2	3	4	VS	S	D	VD
Developed applications execute quickly	1	2	3	4	VS	S	D	VD
Developed applications are platform-independent	1	2	3	4	VS	S	D	VD
Tools are adequately supported	1	2	3	4	VS	S	D	VD
Tools create reliable, high quality applications	1	2	3	4	VS	S	D	VD
Tools provide fast response to developer	1	2	3	4	VS	S	D	VD
Tools run on any platform	1	2	3	4	VS	S	D	VD
Tools are state-of-the-technical-art	1	2	3	4	VS	S	D	VD
Tools have low cost	1	2	3	4	VS	S	D	VD

34. Please indicate *how important* each of the following compiler-related features is to you, and *how satisfied* you are with the C Set/2 compiler IBM has provided.

(1= Extremely Important; 4= Unimportant) (VS= V. Satisfied; VD= V. Dissatisfied)

Ease of installation	1	2	3	4	VS	S	D	VD
Completeness of documentation (online & hardcopy)	1	2	3	4	VS	S	D	VD
Ease of use of documentation (online & hardcopy)	1	2	3	4	VS	S	D	VD
Runtime library support	1	2	3	4	VS	S	D	VD
Standards conformance	1	2	3	4	VS	S	D	VD
Standards enforcement	1	2	3	4	VS	S	D	VD
DLL build support	1	2	3	4	VS	S	D	VD
Application execution speed	1	2	3	4	VS	S	D	VD
Resource utilization during compile	1	2	3	4	VS	S	D	VD
Ease of migration for 16-bit applications	1	2	3	4	VS	S	D	VD
Effective warning/error messages	1	2	3	4	VS	S	D	VD
Compile time	1	2	3	4	VS	S	D	VD
EXE size	1	2	3	4	VS	S	D	VD
Safe optimization	1	2	3	4	VS	S	D	VD

35. Please indicate *how important* each of the following debugger-related features is to you, and *how satisfied* you are with the C Set/2 debugger IBM has provided.

(1= Extremely Important; 4= Unimportant) (VS= V. Satisfied; VD= V. Dissatisfied)

Ease of installation	1	2	3	4	VS	S	D	VD
Completeness of documentation (online & hardcopy)	1	2	3	4	VS	S	D	VD
Ease of use of documentation (online & hardcopy)	1	2	3	4	VS	S	D	VD
Source level debug	1	2	3	4	VS	S	D	VD
Step mode debug	1	2	3	4	VS	S	D	VD
Synchronous/asynchronous debug modes	1	2	3	4	VS	S	D	VD
Point-and-shoot PM interface	1	2	3	4	VS	S	D	VD

36. Please indicate *how important* each of the following environment-related features is to you, and *how satisfied* you are with the WorkFrame/2 solution IBM has provided.

(1= Extremely Important; 4= Unimportant) (VS= V. Satisfied; VD= V. Dissatisfied)

Ease of installation	1	2	3	4	VS	S	D	VD
Completeness of documentation (online & hardcopy)	1	2	3	4	VS	S	D	VD
Ease of use of documentation (online & hardcopy)	1	2	3	4	VS	S	D	VD
Ease of use of development environment	1	2	3	4	VS	S	D	VD
Ease of integrating non-IBM tools	1	2	3	4	VS	S	D	VD

37. Please indicate *how important* each of the following support-related features is to you, and *how satisfied* you are with the support IBM is providing for C Set/2.

(1= Extremely Important; 4= Unimportant) (VS= V. Satisfied; VD= V. Dissatisfied)

Friendliness of support staff	1	2	3	4	VS	S	D	VD
Turnaround time (problem reported until problem fixed)	1	2	3	4	VS	S	D	VD
Quality of fixes (i.e. Did they fix the problem ?)	1	2	3	4	VS	S	D	VD

Just follow the yellow brick road...

Yes, as one of our readers asked, it is possible to write C programs that do not need to be linked with the C Runtime Libraries. Our library group will try not to feel too hurt. You can build this type of application in both C Set/2 and C Set++, but we think it's a little easier using the latter product because of some of our new features. Features in the C part of C Set++, that is.

There are five areas of potential concern in attempting to build code of this nature:

- compiler options to use
- dependencies on the runtime in user code
- dependencies on the runtime in compiler-generated code
- getting the system to call your entry point without going through the runtime
- finding the command-line arguments

First, let's look at the compiler options you should use when compiling code that is not to be linked with a C library:

- /Rn - use the subset library. Well, you're not going to use any, but this option will also remove the C runtime environment.
- /Gn - suppress generation of default libraries. The compiler would normally generate object records that force the linker to include certain libraries, including the C library, in the link. This option will suppress the default library records. To be safe, use the /NOD (NODEFAULTLIBRARIES) option when linking, and explicitly specify any libraries you want to use in the link statement.

Removing dependencies on the runtime from user code is not exactly rocket science - don't call C library functions! Actually, you can call any C library function which is a built-in or intrinsic function. Code for these functions is generated inline, and no call will be made to functions in the C library. Please note that many of the built-in functions will revert to library function calls when your code is compiled with /O- (See the C Set/2 User's Guide, or the C Set++ Programming Guide for details).

You also don't have to worry very much about the dependencies on the runtime in the generated code. Aside from entry point hooks, which we will deal with shortly, code generated by the compiler may only have three dependencies on the C library:

- Some loops of forms similar to:

```
for(i= LOW_BOUND;i+ + ;i< HIGH_BOUND)
  a[i] =  CONST;
```

may be turned into calls to the memset function. As mentioned above calls will be inlined when code is compiled with /O+ . Alternatively, extract memset.obj from dde4sbs.lib using the LIB utility from WorkFrame/2 and link your application with the object.

- When calling 16-bit functions, the generated code will contain a call to __EDCThunkProlog, and a call to __EDC3216 for each 16-bit call. If you wish to make 16-bit calls in code that is not to be linked with the C library then extract the edc3216.obj from dde4sbs.lib, and link your application to the object. The routines have no dependencies on the rest of the C library.
- If your code contains segmented (_Seg16) pointers which are converted to or from flat pointers, the code for those conversions will reference _DosSelToFlat and _DosFlatToSel. These are simply wrapper functions around the Dos32SelToFlat and Dos32FlatToSel functions in os2386.lib. Again the sel_flat object may be extracted from dde4sbs.lib, and your code linked to the extracted object.

  Please note that the Release 1 compiler is sometimes a little over-zealous about referencing __EDCThunkProlog, __EDC3216, _DosSelToFlat, and _DosFlatToSel when not strictly necessary. We believe we've got it right in C Set+ + .

If you look at the assembly listing file for a C file containing the main function, you'll notice that the compiler generates an external reference to _exentry. This is the main entry point to the C library, and if included will force the system to transfer control to it. To get around this problem, simply don't include a main function in the code for your .EXE (I know that's hard to do for veteran C programmers, but trust your C Set team). However, you now have to make the system transfer control to your entry point. Here is where things become a little easier if you are working with C Set+ + . This release contains a new pragma, #pragma entry, which specifies the function to which control should be transferred. For example, to make stephen the function the system calls:

```
#pragma entry(stephen)
int stephen(void)
{
 .
 .
 .
}
```

If you are working with C Set/2, the easiest way to accomplish this task is to write your entry point in assembler, and place the name of

the entry point on the MODEND statement at the end of the assembler file. Thus, if max is your entry point, use:

```
MODEND max
```

Now, what if you're building a DLL? If you take a look at an assembly listing file for a source file compiled for inclusion in a DLL (with /Ge-), you'll notice an external reference to _dllentry. This routine is the entry point for the DLL, corresponding to _exeentry for .EXEs. You can replace the library version with your own entry point in the same ways we discussed for replacing _exeentry. In your version, be sure to include a definition of _dllentry to avoid the library copy being included. You will also need to provide your own _DLL_InitTerm routine which performs DLL initialization and termination. The C Set/2 User's Guide, and the C Set++ Programmer's Guide contain instructions for writing your own DLL initialization and termination. To avoid any library references, be sure not to call the _CRT_init and _CRT_term functions in _DLL_InitTerm.

Finally, one of the many functions of the C library is to decode the command-line arguments, and place them in argc, and argv. If you are not linking to the C library, and need to get at the command-line, then you can write your own routine to do so. The command line consists of two contiguous strings, the first is the program name, the second the program arguments. A pointer to these strings resides in the Process Information Block (PIB). Here's a sample of how to access them:

```
#define INCL_DOSPROCESS
#include < os2.h>
#include < stdio.h>
#include < string.h>

int main(void)
{
   PPIB pPib;   /* pointer to process information block    */
   PTIB pTib;   /* pointer to thread information block     */
   char *args;  /* pointer to string containing arguments */

   DosGetInfoBlocks(&pTib, &pPib);
   args =  pPib-> pib_pchcmd +  strlen(pPib-> pib_pchcmd) +  1;

   printf("Program Name: % s  Args: % s\n", pPib-> pib_pchcmd, args);

   return 0;
}
```

Unfortunately, parsing the arguments is up to you!

I'm sorry, I'll say that again....

Since we published the November newsletter, we have removed one of the debugging options from the C++ beta compiler. The /Ts option is no longer supported but, don't worry, we haven't left you in the lurch: the support this option provided (allowing the debugger to maintain the call stack across functions that don't chain EBP) is still there, but "option-free", i.e. it's now implemented directly in IPMD.

In our last edition, we ran a Q&A about malloc and the answer we gave was:

> You can use either malloc or DosAllocMem to allocate storage. Use malloc if you plan to reallocate the storage with realloc. Otherwise, use DosAllocMem for better performance.

We may have inadvertently made it sound as if there are problems with malloc. There aren't. malloc is general purpose to avoid worst case behaviour, which might result in a performance trade-off. If performance is important, then DosAllocMem would suit you better, but remember, using it loses you portability.

More queries about the November 92 Q&A

Question	Answer
Q: /Gr- and ring 0 I didn't know I could put my code at ring 0 simply by saying /Gr-. I can't believe that if you say /Gr- the code will run at ring 0 and the performance will suffer? Does this mean that the libraries (such as printf) and the application library can be used from ring? Please clarify.	We should have said ***can be run***. Performance could suffer because we disable optimizations that are unsafe at ring 0. It's used when writing VDDs.
Q: Subsystem library I always say to use the applications libraries because they are full function, that all routines are reentrant if you use the multi-thread libraries, and because exceptions are handled by the library. The 11/92 newsletter said to use the subsystem library. I don't think that's good advice - am I confused?	We think it ***is*** good advice - all the functions in the subsystem library are reentrant except the malloc family. Using the subsystem library wherever possible will reduce your exe size and initialization time.
Q: Double-word alignment The 486 likes quad-word alignment. Shouldn't the documentation specify align 8, and the compile emit code which says align 8?	When we generate code for the 486 (G4 option), we align functions and labels on 16 byte boundaries. In C Set++, we are aligning doubles and long doubles on 8 byte boundaries. We think this gives you better performance.

Question	Answer
Q: /ALIGN is not a good idea The November compiler newsletter states: If you link your program in a separate link step, specify the /ALIGN:1 linker option to align segments on 1-byte boundaries. The default alignment is 4-byte boundaries. I don't think this is a good idea - it slows down references to memory as it knocks them out of alignment, or do I not understand?	We think it is ok - /ALIGN only affects alignment on DISK.
Q: func16 - why do it this way? If the object is to eliminate the pragma and in-line, why pose the examples you did?	We ***are*** recommending moving away from #pragma linkage to linkage keywords. However, for completeness, we showed both ways of declaring a linkage convention on a function. The purpose of our last sample on page 17 was to demonstrate how to set the linkage convention on a typedef of a function.
Q: /G3 - why the extra code? Why does optimizing for the 486 produce extra code?	Several reasons. More alignment - we align on 16 byte boundaries instead of 4; several longer instruction sequences are more efficient on the 486.
Q: DDE4.MSG I wasn't aware that DDE4.MSG should be bound into an .EXE which will be run on a hard disk which doesn't have the compiler installed. Please elaborate.	DDE4.MSG is the runtime message file. Since running your application may cause runtime messages to be issued, if this file is not so bound, the result may be the infamous "Message file not found" message. This is worse than it sounds because it will occur on your customer's machine when he's running your application!

You ask, we answer

These questions are culled from a variety of sources: from internal and external fora; from customer calls; and from the reply forms mailed in by our readers.

As some questions are posed by more than one user, you may notice "your" inquiry worded slightly differently. We've probably consolidated similar queries that can share a detailed answer. If you feel we haven't quite got your point, let us know. As you can see by the above section, we're happy to respond to comments on our answers. And now for the new stuff:

Question	Answer
Q: CSD levels How do I know which was the latest CSD applied to my C Set/2 ? Should I install the previous fixes, or may I apply CSD nn (whatever is the latest level)?	You can determine the current CSD level of C Set/2 (and everything else in your system) by typing SYSLEVEL at the OS/2 prompt. The CSD's are cumulative - the latest will completely update you. BUT!!! See the next Q&A.
Q: C Set/2 traps I can't get the latest C Set/2 to work - I installed the beta and applied the CSDs but it just traps.	No, No, No! Don't ever try applying a CSD (which is for Version 1, the currently commercially available product) to beta code (which is Version 2). The beta code we release is as up to date as we can make it. We may issue a new beta to fix problems, but we will never have a CSD for beta code.
Q: Using /NOI I understand about using /NOI when linking both a referenced DLL and a referencing EXE or DLL, but do you also have to specify /NOI on the IMPLIB command when building the import library for a referenced DLL? I get by without doing this, but wonder if I've just been lucky.	Yes, you need to use /NOI when building import libs for C/C++ code.
Q: EDC0185 on _beginthread Why am I getting this error: EDC0185: Function _beginthread has not been prototyped prior to use. when I do have stdlib.h included and I am using /Gm+ in the compile?	Start by compiling with -p and look at the .i file that is produced. Is there a prototype for _beginthread in the .i file? If not, look a little more carefully at your code to make sure you're not conditionally compiling it out or something. What other compile flags are you using besides /Gm+ ? You can't use /Sa or /S2. They both cause the _beginthread prototype to be excluded from the compile in stdlib.h. The language level must be Extended to use the thread functions.
Q: OS/2 1.3 Compatibility I'm new to 2.0 and have many tools under OS/2 1.3. Can I use either the C Set/2 or C Set++ to build code for those platforms? Is it a matter of using perhaps #pragma linkage statements to force 16 bit entry points and then using a 16 bit linker?	Sorry. Our code will not run on OS/2 1.X. The 16-bit entry points all thunk to 32-bit on entry. Essentially we only generate 32-bit code.

Question	Answer
Q: Link failed: symbol defined more than once This module linked before I applied the AT&T Basic Class Library headers I obtained separately. Now I keep getting the following errors. What caused this? ICC.EXE @f:\nm000372. Operating System/2 Linear Executable Linker Version 2.01.004 Mar 09 1993 Copyright (C) IBM Corporation 1988-1993. Copyright (C) Microsoft Corp. 1988-1993. All rights reserved. Object Modules [.obj] /de /pmtype:pm /noi /align:16 /exepack /base:65536 + Object Modules [.obj]: "tmsentry.OBJ" + Object Modules [.obj]: ".\tempinc\iavlkss.obj" + Object Modules [.obj]: ".\tempinc\isskss.obj" + Object Modules [.obj]: "f:\c0176AAA.obj" Run File [tmsentry.exe[: "TMS.EXE" List File [nul.map]: Libraries [.lib]: "ibase.LIB" + Libraries [.lib]: "ibaseapp.LIB" + Libraries [.lib]: "ibasectl.LIB" + Libraries [.lib]: "iclcc.lib" + Libraries [.lib]: "idrag.lib" + Libraries [.lib]: "icnr.LIB" Definitions File [nul.def]: TMS.DEF" X:\IBMCPP\LIB\DDE4SBS.LIB(CTYPETBL.C) : error L2025: _ctype : symbol defined more than once X:\IBMCPP\LIB\DDE4SBS.LIB(EDCINIT.C) : error L2025: _tzname : symbol defined more than once X:\IBMCPP\LIB\DDE4SBS.LIB(EDCINIT.C) : error L2025: _osmajor : symbol defined more than once X:\IBMCPP\LIB\DDE4SBS.LIB(EDCINIT.C) : error L2025: _osminor : symbol defined more than once X:\IBMCPP\LIB\DDE4SBS.LIB(EDCINIT.C) : error L2025: _environ : symbol defined more than once X:\IBMCPP\LIB\DDE4SBS.LIB(EDCINIT.C) : error L2025: errno : symbol defined more than once X:\IBMCPP\LIB\DDE4SBS.LIB(EDCINIT.C) : error L2025: _doserrno : symbol defined more than once X:\IBMCPP\LIB\DDE4SBS.LIB(H:\OS2\STREAM.C) : error L2025: ios::stdioflush : symbol defined more than once X:\IBMCPP\LIB\DDE4SBS.LIB(H:\OS2\STREAM.C) : error L2025: ios::nextword : symbol defined more than once There were 9 errors detected	Try to link your program using library in this order: 1. icnr.lib 2. idrag.lib 3. ibasectl.lib 4. ibaseapp.lib 5. ibase.lib 6. iclcc.lib This should solve your problem.
Q: Creating and using a DLL I have gotten to the point where I can successfully create a DLL and use it on my machine. When I attempt to take my EXE and DLL to another machine (that doesn't have the toolkit, compiler, etc. installed) I get a SYS1804 error message indicating that the file DDE4SBS cannot be found. What do I need to do when creating my DLL so that I do not need this file?	Don't compile the modules with /Gd. Either that, or copy DDE4SBS.DLL over to the other machine too.

Question	Answer
Q: OPTLINK problems We are currently converting code from Zortech to C/C++ beta. Zortech had no problems, but the beta code does, and we don't know why. Can you help? It is about one line of code : signal(iSigval, SIG_DFL); It gives the next errors : error SYS3055: "void(*)(int)" cannot be converted to "extern "OPTLINK" void(*)(int)". informational SYS3306: The previous message applies to argument 2 of function "signal(int,extern "OPTLINK" void(*)(int))". It looks if the signal prototype and the SIG_DFL define are not of the same type. But if I look at the signal header file, I see : #define SIG_DFL ((void (*)(int))0) /* def signal actn */ void (* _Optlink signal(int, void (*)(int)))(int) So, it looks alright. Can somebody give me a hint ?	Try declaring iSigVal as an _Optlink function.
Q: /Ms forces System linkage except... The /Ms compiler option forces system linkage except for functions wrappered in extern "C". The workaround is to specify: extern "C" < return type> _System < function> Is this right? If so, where is this in the documentation? Without this workaround applied, there are no link errors, and the program runs to the point of the CPP routine returning to the C routine. This is where different things can happen. In our case, part of an address space is being corrupted.	No, this is not working as designed. Specifying /Ms on the command line should generate System linkage for functions wrapped in extern "C" also. This is fixed but unfortunately, the change did not make it into the Beta.
Q: C++ and WorkFrame/2 Does C++ integrate in the WorkFrame environment like C Set/2? 1. Can I specify the compiler/linker options for a project using the windows like I can with C Set/2? 2. Can I use the Tools pull down Make file utility to generate a make file that can be used to compiler/link a c++ program? 3. When running a compiler, can I "click" on a compiler error message in the little window and get *popped* into EPM at the line where the error occurs? 4. Will the Compiler on the EPM action bar be available, so I can jump from error to error and get help describing what is wrong?	1. Yes. 2. Yes 3. Yes 4. yes The only thing that you will have to do is to create a language profile for C++ . There is documentation on how to do this in the WorkFrame documentation.

Question	Answer

Q: __strcmp and sys_errlist
 	
Can someone tell me why strcmp is mapped to __strcmp via a macro? Some C++ code I am trying to port defines a strcmp friend function and the compile appears to be getting the macro expansion. For example:
 	
friend int strcmp(const String x, const String y);
becomes...
friend int __strcmp((const String x), (const String y));
 	
The compiler does not like the second left paren after __strcmp. Does anyone know of a way to override this? If I change the source to __strcmp then I get a clean compile (don't want to mess with it if I don't have to).
	To protect your function from the macro, you can add extra () around the name of the function so that the preprocessor doesn't get in the way. For the final product, we are fixing the header files so that this will not happen.

Q: IPFC. Where can I find out about the tags I need to use for an on-line help book to be compiled using the IPFC ?? There isn't an .INF version of this is there, or a softcopy ??	Information Presentation Facility Guide and Reference, S10G-6262. Yes there is one IPFC20.INF file shipped with OS/2 Toolkit 2.0.
Q: WIBNI include nest were printed on error I am converting from ZTC++ to C++ Beta and I am getting errors in the includes that C++ Beta does not like but which work in ZTC++. The errors usually cause the next include to generate the error message so it is difficult to determine where the error actually is. These includes are nested and use the #ifndef model to prevent duplicates. It would be much easier if the "include stack" were printed at an error. Then I could determine what the previous include was.	Preprocessing your program to a file or to the screen with the '/P' or '/Pd' options is one way that might help, but problem with /Pd is that it displays the contents of each include file. With programs that use ICLCC, that's an awful lot of scrolling and waiting. You could redirect to file and edit it but we need something more useful like Zortech's -v switch. Noted as a requirement for the next release...
Q: Shipping online documentation .boo vs. .inf Why are there no.BOO files on the CD-ROM?	For this release, we are providing a number of books (but not all) in .INF format, and the rest in hardcopy. No BookManager is shipped with the product. Customers *will* however be able to purchase one of 2 CD-ROMs with all of our books (and more!) on them in BookManager format, with a gratis copy of the Library Reader included so they can read them without having to buy the READ product.

Question	Answer
Q: Problem with "MAKE" I'm a novice and have just installed WorkFrame/2, CSET/2, and the Developer's Toolkit. In stepping through the tutorial, I was able to edit the sample C program and to set up a MAKE file. When I try to do the "make", I get the following error: oICC.EXE /Ti /C .\HELLO2.c SYS1041: The name specified is not recognized as an internal or external command, operable program or batch file. NMAKE: fatal error U1077: 'C:\OS2\CMD.EXE' : return code '17' Also, at an earlier point in the tutorial (I believe it was when I selected CSET/2 as my language) I got an error message about being unable to access "help" files. I suspect that something is wrong with my CONFIG.SYS, but I am not sure what. I am also new to OS/2. Any ideas on what I might have done wrong ?	Did you select "Update your config.sys" when you installed C Set/2? If not, you should have a file csetenv.cmd in your \ibmc\bin directory. You need to either include the environment variables from there into your config.sys, or run the .cmd file before you try and execute icc. If you choose the later, the best approach is to run it as you start WorkFrame - add it to the end of your wfenv.cmd, and start WorkFrame via that.
Q: WF/2 1.1 You have added WorkFrame/2 1.1 to the latest beta drop. What's the added value of WF/2 1.1 compared to WF/2 1.0?	Version 1.1 is, for the most part, the same as version 1.0. Changes have gone in to support intermediate code file generation (in the makefile creation utility and other miscellaneous places), but that's probably the only visible change. Other changes (internally) include bug fixes, moving some code (LIB) to 32 bit, and improved DBCS support.
Q: Copy a project. Is there a way to copy a (generic) project to make a new project by Drag/Drop method?	Yes; from the project control, select new. You can now drag an existing project from the project control, into the new project dialog.
Q: Make file Does anyone know if WF/2 will remake the project if ONLY the make file itself is changed?	The makefile creation utility places a dependency for every file on the makefile itself, which means that if you change the file, everything will be rebuilt. Note that if you only change the options (from within WorkFrame), the makefile will not be changed -- you'll need to re-run the makefile creation utility.
Q: Document S10G-4475 Document 10G4449 refers to S10G-4475 'IBM WorkFrame/2 Introduction' What is the value of this document compared with the Tutorial?	The only thing the WF/2 Introduction booklet (S10G-4475) covers that you wouldn't get from the tutorial (10G4449) is installation

Question	Answer
Q: Customizing Workframe/2 I am a novice at Workframe/2 and have two problems: • How can I make my editor of choice (EPM) a child of the Workframe window? (When I go to edit, it gets placed on top of work-frame and then when I switch back to work-frame, the editor gets thrown back into the background. My ideal is to keep the editor open and part of the environment.) • There's supposed to be contextual help when using the EPM editor, but I can not seem to get this working, it is saying that some file is not found.	• Go to Configure > preferences, and select the 'capture tools' button. Alternatively, you may: – Size the WorkFrame/2 Window so that it is just as wide as the desktop but only as high as the menu bar roughly 1/4 inch tall. You'll now be able to see your editor behind WorkFrame. – Put ?:\TOOLKT20\BOOK in DPATH and reboot (where ? is your boot drive)
Q: Installing Workframe/2 on a LAN server I am attempting to install Workframe/2 on a LAN server so everyone in my department does not have to do an install, saving disk space and time (I plan to do the same with C Set/2 and Toolkit 2.0). I had Workframe/2 installed on my D: drive, and erased the directory after installing it on the LAN (my L: drive). I updated my CON-FIG.SYS file to point to the directories on the LAN. When I try to start Workframe/2 I get error messages saying it cannot load the IPF help file and cannot load the Workframe profile. What can I do to resolve this? Also, when I had Workframe/2 installed on my D: drive, the project files were stored in the D:\IBMWF direc-tory. I want those types of files to be on each user's disk. What do I need to do to make that happen as well. Thanks.	Both you and your users should run the ADDUSER command -- that will setup your environment so WorkFrame can find the various files. The project files (etc) will be stored in the target directory, specified when you run ADDUSER.
Q: Source Code I am a novice user of the Workframe, I am trying to view the source code in the debugger, but always get the unassembled code instead. I followed the steps in the Workframe tutorial, I though this would solve the problem, but I am still getting the unassembled code. Could you please tell me what I am doing wrong.	Make sure that you specified the debug options for compile and link when you do a makefile create. Also, if you're using C Set+ + , rather than C Set/2, check that you are using the right linker - the one that comes with the beta code.

Question	Answer
Q: Printing From Action/Monitor Windows to IBM 4216 Does anyone know how workframe handles printing? There is essentially no documentation either hardcopy or in help that I can find. In particular, I want to press the print buttons on the monitor or action log windows and get output on my 4216-031 which is set up through AES4216 for port emulation switching as follows: Proprinter on LPT1:, HP LaserJet on LPT2:, and PostScript on LPT3:. When I press the print buttons on the two windows in question, I see the job go through the LPT1: print queue and the red light blinks on the 4216 for a few seconds..but nothing comes out. I use all three ports extensively in a number of other applications with no problem. Does anyone know how to port the print output to a file?	WorkFrame will currently only print to text printers. If you wish to print to a PostScript printer, use the save button onstead, and then print the resulting file.
Q: How to pass a piped in file as a runtime argument How can I pass in an argument which is being piped into the exe, when using IPMD? For example, the command line version of running the exe is: MyExe < myfile How can I pass in myfile to IPMD to have the same effect?	This should work if you start IPMD without a debuggee name (to get to the startup dialog) and then type the "< filename" in the parameters field.
Q: SOURCE FILES Is there a way to say to IPMD "Please look in a specific directory for source files." ? All the DLLs that I develop are stored in < dllname> \SRC whereas test programs which use them are in < dllname> \TEST. I run IPMD from the latter directory, but IPMD cannot find there the .C files, of course. So he asks me where files are (more than one) and I must type ..\SRC each time. I'd like to save such information once for ever. Is that possible? I don't want to mix DEVELOPMENT and TEST environments...	Use the environment variable PMDPATH to specify a path to search for the source files, i.e. "SET PMDPATH= D:\SOURCE1;D:\SOURCE2;D:\SOURCE3;").

Question	Answer
Q: IPMD WIBNI: jump to a statement I would like to have the possibility in the debugger to continue execution at an arbitrary point, instead of following the natural program flow all the time. This comes in very handy if you want to skip a certain function call, for example. Any chance of ever getting such a function, is it already there, or is this considered to be too big a gun to shoot yourself in the foot with?	It's already there in the release 2 version of IPMD (Jump to location), which is part of the beta code.
Q: White vertical strip Sometimes, when I debug a DLL, I get a WHITE VERTICAL STRIP about 15 characters wide that clears the left side of the source code window. I checked that SOURCE and EXECUTABLE were in sync, but I still get that strip. I can't debug. I can't see the code. Why?	This is caused by a display driver/font problem (in the operating system). Try changing the font and/or the font size for the window.
Q: Can't Set Breakpoints in Source View Can anyone suggest a reason why I can't set any form of breakpoints in source view. I can set them in dissassambly view. Is this a bug? I have tried setting breaks by double-clicks on the line prefix as well as with the menu options. The double-click has no effect and the menu option reports "error setting breakpoint".	You have to make sure that you set the breakpoint by double-clicking in the PREFIX area of an EXECUTABLE in the program view windows. If the statement is not executable (usually, they have a different color), you won't be able to set the breakpoint there. Be aware that most people who have this problem have used the wrong linker.
Q: Minimized Work Window Every time I do a compile or Make, the work window is started minimized. I remember it coming up 'normal' in some previous life, so I must have done something to cause this. Can someone tell me what.	Under Configure-> Preferences, deselect "Minimize Monitor". Don't forget to save the options (File-> Save Settings-> Profile).
Q: Paint Update Rectangle Once into the PaintHandler routine, after retrieving the hps using presSpace(), how does one obtain the update rectangle? This is normal procedure so that one does not draw more than is necessary to repair the window.	Try IPaintEvent::rect() and IPaintEvent::exposedRect().
Q: Macro preprocessor The macro preprocessor facility of the compiler is rudimentary. When there is a need to initialize a large number of complex data structures, it is very cumbersome. It is nowhere near that of Assembler H.	We appreciate your need for additional function in the preprocessor, but as we are limited to the capabilities of the C language itself, you'll have to settle for less than the macro facilities in 370 Assembler H.

Question	Answer
Q: Usability problem Whenever I return from debugging a function, the changes that I applied to the layout of the LOCAL VARIABLES list disappear, and I get the default layout. That's very annoying. Here's an example. I'm debugging function X. I open the LOCAL VARIABLES window and I click on a certain pointer to see the pointed structure. In such structure I change the representation of two strings and three pointers. Then I shrink a window that shows a structure in which I'm not interested, and enlarge another window with a very long structure. Now the LOCAL VARS window is OK, and I proceed debugging. I step a couple of times and then enter in function Y, called by X. I debug Y and return. In the meantime I never closed the LOCAL VARS window. As I return in X *ALL* the changes applied to the LOCAL VARS window go away, and I have to re-do everything. Any way to avoid that?	There is no way to do exactly what you want. The best way to come close to it is to place the variables you are interested in into a private or a program monitor list. The format of these windows is remembered. In release 2 of the debugger, you can set the default representations for variables in the "Default data representation" dialogs.
Q: Viewing function I'm currently using the latest version of IPMD and am running a program through IPMD to the point where I get a message: OS/2 Application Exception occurred: Access violation. The only push button available is: Examine/Retry The function that I'm receiving this from is a Disassembled strncpy function. My code has several of these. My question is how do I, if possible, go back to the actual code where the strncpy function is located. The strncpy function is located in one of my DLL's.	Use the stack window and double click on the function that called strncpy().
Q: Posting Events to a Message Queue I am trying to implement a single server thread to handle and serialize operations from the PM thread. Supposedly I can postToQueue an event to this thread's queue. How do I connect event handles in the thread when it has no windows? Or do I just do it at the OS2 level on the thread side? These events are IThreadMemberFns to be executed by the thread. This thread will also get timer "events" if that makes any difference.	Hmmm, I'm afraid that's a scenario we didn't design for. The best solution is to create an IObjectWindow in the thread and attach handlers to it. This would allow the use of the dispatching scheme with very little window overhead.

Question	Answer

Q: What happens when using a handler declared as IHandler *
 	
Does the appropriate event handling function get called if my class contains a pointer to an IHandler. In other words, if I have a window class and a command handler class like so:
 	
class MyWindow : public IDialogWindow {
IHandler *p;
public:
MyWindow(IHandler *x) : p(x) {
p-> handlerEventsFor(this);
//...
}
//...
};
class MyCmdHandler : public ICommandHandler {
//...
Boolean command(ICmdEvent&);
};
 	
void main() {
MyCmdHandler mycmdhandler;
MyWindow mywindow(mycmdhandler);
//...
}
 	
will MyCmdHandler::command() get called for WM_COMMAND or will things go haywire? I assume it doesn't get called since command() is not part of the interface of IHandler. I just want to be sure.
	Your handler will work just fine. The message dispatching code calls the handler's dispatchHandlerEvent function. In your case, this will be the implementation you inherit from ICommandHandler. This function will call this-> command(), which you've re-implemented and thus will invoke *your* command() function. This is the "raison d'etre" of C+ + virtual functions.

Q: rightJustify() The global version of rightJustify() seems to have disappeared. When compiling the following code: = = = =

Question	Answer
Q: Link error I am compiling a program with the following options icc /C /Gm /W2 /Ti test.cpp I then link the program as follows link386 test.obj /pm:pm /noi /de,,,os2386 dde4mbs dde4muii iclcc On the link, I receive a whole bunch of 'invalid format of debugging information'. I am using the lastest beta drop and have applied the beta4fix.ram and tlkt20ud.ram files. Any ideas?	This looks like what happens when you use an old version of link386. Make sure the one that comes with the beta gets used.
Q: Deferred creation of dialog using IFrame & deferCreation We are trying to defer creation of a dialog by inheriting from IFrame and passing the Style of deferCreation to the CTOR of IFrame. We then used tryToLoadDlg() in our class CTOR to create the dialog. This worked except that we appeared to not be receiving any events such as command(). All worked fine if we simply passed the dialog construction parms to the IFrame in the CTOR (the normal situation). Firstly, is it correct to use tryToLoadDlg to create out window from a dialog template when deferring? Secondly, why this prevent us from getting command events? We are linking to the dde4muii dll rather than the static lib's if this is of any help.	IFrameWindow::tryToLoadDialog() and IFrameWindow::initialize() both call IFrameWindow::start() when and if the window is created. start() adds as a handler the "default" frame handler. This is an instance of class IFrameHandler and has no relationship to any other handlers you may have lying around. So, my questions are: what object's command() function is it you expect to have called? Is this handler object added to the frame? Is this handler added before or after the call to tryToLoadDialog()? My suspicion is that either your handler is never added, or, is trumped by the default frame handler that is added subsequently. The default frame handler will swallow WM_COMMANDs if you don't have a client window (which is usually the case for dialog windows). If either of these is the case, then simply add your handler *after* calling tryToLoadDialog(). Alternatively, if you have a subclass of IFrameHandler you could override addDefaultHandler() and add your handler at that point.
Q: First item of popup menu automatically selected? I have a program that displays a popup menu. When I click mouse button 2, the popup menu appears and then my overridden 'menuSelected' function gets called automatically (when no menu item was selected) and the case for the first menu item is executed. Is this a known bug?	menuSelected() is called when the menu item has the "selection cursor" which is different than a menu item being clicked on. I believe the first menu item receives the selection cursor when the menu is activated. When a menu item is clicked on you'll get a WM_COMMAND (aka command() called in an ICommandHandler). Does this account for what you're seeing?

Stamping out a bad habit

A thank you from your editor....for all of you who've mailed cards, letters, and the reader's form back to us...thanks for the lovely collection of stamps, from Finland to South Africa, New Zealand to Sweden...they're saved and glued to a board. Down with franking, long live the stamp!

Toronto strikes again

Our cousins down the road, the Data Base folks, have begun issuing a newsletter, and we're pleased to bring you an advertisement on their behalf.

IBM DATABASE 2 OS/2 Version 1 (DB2/2) and IBM DATABASE 2 AIX/6000 Version 1 (DB2/6000) are relational database management systems which extend IBM's SAA relational database technology to the single-user workstation and client/server LAN environment and are a key component in IBM's Information Warehouse (TM) framework.

DDCS/2 Version 2 and DDCS/6000 Version 1 provide the ability to develop and run application programs that connect to, access, and update host databases from OS/2, AIX, DOS and DOS Windows (1) database client workstations.

To receive the **DB2 Newsletter**, mail or FAX a note to:

IBM Canada Ltd
Database Technology Planning
c/o Lou Pedron
31/110/895/TOR
895 Don Mills Road
North York, Ontario
Canada M3C 1W3

FAX 416-448-2114

A word from your editor

If you didn't receive this newsletter mailed direct from IBM, and you would like regular hardcopy, then let us know. Mail in the reply form at the back with your request to join us, plus your full mailing address, and we'll add you to our mailing list, wherever you are. Yes, wherever you are...from Argentina to Zaire, if you have a mail service, we'll mail you our newsletter.

For the many of you sending in the reply forms with your comments, we've often had the need to call you to discuss your comments further. It's a great help if you include your phone number. Thanks!

This newsletter is available softcopy on several networks. If you obtained your copy electronically, you may not be on the C Set + + mailing list. If you're not, you're missing out! You're not receiving the product information, information updates, and other general mailouts that others are. You can be on our list for these goodies, with or without hardcopy newsletter distribution. Just mail/fax your full address (phone number is handy too) using the Reader's Reply Form at the back of this copy, and circle whether you want the NL hardcopy, or just be added to the mailing list.

This newsletter was produced by the OS/2 C/C+ + Planning department of the IBM Canada Toronto Lab. For further information on any of the products mentioned, please contact your local IBM office, or an authorized IBM Business Partner.

Numerous product references in this publication are registered trademarks or trademarks of International Business Machines Corporation. IBM Canada Ltd., a related company, is a registered user.

Sample code is provided for information purposes only, and is used by readers at their risk. IBM makes an effort to provide accurate and safe code examples but does not warrant their correctness.

This newsletter was created and marked for processing using IBM BookMaster (Program Number 5688-015) and IBM Document Composition Facility (DCF)™ (Program Number 5748-XX9).

The final copy was printed on an IBM 3825 Page Printer, an Advanced Function Printer.

This newsletter is ê IBM Corporation 1993.

IBM

OS/2 Compiler News No.4 April 93 issue

Reader's Reply Form

1. *Did you find this newsletter useful?*
2. *Is there any way you think we could improve this newsletter?*
3. *Is there any compiler-related subject you would like to see addressed in this newsletter?*
4. *If you received your copy electronically, would you like (circle one):*
   - *hardcopy mailed to you?*
   - *mailing list only, no hardcopy mailing?*

Please note:

- *IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you, and all such information will be considered non-confidential.*
- *Do not use this form to report compiler problems or to request copies of publications. Instead, contact your IBM representative or an authorised IBM Business Partner.*
- *If you wish, you may include your name, address, and company name if applicable, and your phone number.*

_____

_____

_____

_____

_____

Thank you for your cooperation and help. You can either mail this form to us, or hand it into an IBM office for forwarding.

You can also fax the form to us. Our fax is 416-448-6057. Please mark your fax for the attention of MJ Houghton.

OS/2 Compiler News No.4 April 93 issue

Reader's Reply Form

Fold here and tape.....fold here and tape.....fold here and tape.....fold here and tape.....fold here and tape..... fold

IBM

IBM Canada Ltd
Workstation Languages Planning
Dept 394
22/394/844/TOR
844 Don Mills Road
North York
Ontario, M3C 1V7
Canada

Fold here and tape.....fold here and tape.....fold here and tape.....fold here and tape.....fold here and tape.....fold