Interactive Document Compiler

Version 1.36
Sampler 1.0

December 1993

CONTENTS

OVERVIEW

The Interactive Document Compiler is a system for the rapid production of interactive documents in Asymmetrix Toolbook 1.53 running under Windows. The documents can range from simple presentations to more complex computer-based learning applications. A document produced using IDC consists of a title page and a number of pages which have been created according to the formats attached to the compiler. IDC itself is a Toolbook application and each format is a Toolbook page with background objects and scripts. During compiling format pages are replicated and filled out using text and images specified in a source document. The source document is a standard text file (such as might be created with Notepad or a word processor and saved in plain text form) marked up with tags which conform to the Standard Generalized Markup Language (SGML). Each format has its own tags and syntax, but different formats can be intermixed in a single source document, provided that the compiler used has all the relevant formats attached. IDC includes support for hypertext, links to other interactive documents and the ability to launch other Windows applications asynchronously.

IDC is consistent with an authoring methodology which separates as far as possible content, presentation and interaction. Presentation and interaction are provided independently: the first through the objects and the second through their scripts. A format combines presentation with interaction, but is content-free. The structured content is provided by the author through the marked up text file and the images.

It should be noted that there is no such thing as *the* Interactive Document Compiler rather there are a number of such compilers each with a set of interchangeable formats. This sampler provides one IDC which includes a number of illustrative formats, some source documents and the compiled interactive documents corresponding to those documents. It is possible to create your own documents with this compiler, or amend the source documents and recompile in order that some appreciation of IDC may be obtained. However, it is not possible to change any of the formats though they may be extracted from, removed from or (re-)attached to the compiler. The full distribution of IDC (should there be interst in it) will allow the modification of existing formats and the creation of new formats. The modification and design of formats requires a throrough knowledge of Toolbook including the Openscript language. Anything that is possible in Toolbook should be possible within a format. The advantage of the IDC approach is that only Designers rather than Authors need to be competent in Toolbook. It should also be noted that if something cannot be achieved with Toolbook then likewise for IDC. IDC adds no intrinsic functionality to Toolbook, it merely provides a systematic methodology for its use.

IDC runs under Windows and needs the full authoring version of Toolbook 1.53, but to run the compiled interactive documents it is only necessary to have the run-time version of Toolbook. The licensing agreement that we have for Toolbook does not allow us to distribute either of these with IDC excepting the run-time system within the University of Leeds.

The sample interactive document (SAMPLER.TBK) can be used without knowing anything further provided that TBOOK.EXE and its component files are available on your system and the installation procedure described in README.TXT has been carried out.

IDC has been developed by:

Kenneth Tait,
Computer Based Learning Unit
The University of Leeds
Leeds, LS2 9JT, UK
Tel: +44 532 334628
Fax: +44 532 334635
Email: ken@uk.ac.leeds.cbl
ken@cbl.leeds.ac.uk

THE SAMPLER

The Sampler Package consists of a Toolbook application produced using IDC, the IDC that produced it, the source document and a numberof other files either required by the sampler or which can be used with IDC as described in the section SOME THINGS TO TRY. The IDC has seven formats (plus an extra one to be used in one of the suggested exercises) and can be used to create new Toolbook applications from source documents written to match the syntax of one or more of the formats in the IDC. Each of the formats in the IDC Sampler will be described giving an outline of the interaction supported and a brief description of the syntax to be used in the marked up source document, but before that we will describe the structure of a complete source document. The reader should keep in mind that this document is selective and does not contain everything about IDC and its use.

Structure of a Source Document

The structure of a source document for IDC is :

<header start=*first*>
<title>*Text of title*</title>
<credits>*Information about authors and contributers*</credits>
<date>*Text giving date*</date>
<doc>*Any informaion about the development and status of the document*</doc>
<caption>*Text shown in the title bar of the window displaying the document*</caption>
</header>
<body>

Pages marked up to be compatible with specific formats
</body>

We shall adopt the convention that italicised words an phrases are descriptive of what should be present. The three character strings '<', '>' and '</' are part of the syntax that we use and correspond to the SGML character strings called **stago** (Start-TAG Open), **tagc** (TAG Close) and **etago** (end-TAG Open). In principle SGML allows any strings to be used for these purposes, but we will use those used in the above example, and which form part of the SGML reference syntax. The part of a document between a start-tag and an end-tag is the content of the tag. The case of the characters in the tags is not significant, in fact case is only significant in text that will be displayed.

In SGML tags may have attributes. An attribute is given a value. In the above example 'start' is an attribute and first is its value. For an interactive document the value of the start attribute (which is compulsory in an IDC source document) is the name of the first page to be presented to the reader. A tag may have several attributes; the declarations of these attributes must be separated from the name of the tag and any other declarations by a space; there must be no spaces at either side of the equals sign. Values can be any sequence of characters excluding '<', '=', and '>'. Values of attributes which include commas or spaces must be enclosed in quotation marks (for example list="1,2,3" and name="Charles Darwin"). Putting quotation marks around a value which contains no commas or spaces is optional.

Line breaks have no significance at all, but it is sensible to place them just after or just before tags.

A tag of the form:

<!-- *text* -->

is a comment and is totally ignored. It is also possible to include an element of the form:

-- *text* --

in any tag as a comment. (Note that the proper use of spaces is important.)

The content of each of the five tags used in the header (title, credits, doc, caption, copyright and date) is text, and the sequence in which they appear is irrelevant. (The contents of credits and doc may be marked up text as described for the Hyperbook format but without anchors.) The information in the header (except the content of doc and caption) is used on the title page of the document .

It is not necessary for the header to have any content at all and so when creating a source document it is quite acceptable to begin it with

<header start=*first*></header>

where *first* is replaced by the name of the page to be displayed first.

Each page must be marked up to correspond to a format which is able to intepret that markup. In identifying a page we do not link it to a particular format but mark it as a page of type f0, f1,..., f9. When we come to compile the document we associate an actual format with f0, f1, ... as necessary. This allows us in a particular interactive document to, for example, identify, say, pages which have the same markup as different in purpose and then experiment with which of several compatible formats we would like to use to interpret these pages. In the examples in this document we will occasionally use <f?> and </f?> as the start-tag and end-tag of a page, though when marking up the document the author must replace the question marks by a digit. All pages must have names which are unique within the document. The name is assigned through the name attribute as follows:

> <f?> name=*uniquename*>*content*</f?>

In addition to pages the author can insert remarks which have the form:

> <fn name=*uniquename*>*text*</fn>

These are used (for example) by the Hyperbook format to create pop-ups.

In SGML, with which IDC markup is consistent, it is possible (among other things) to declare entities which are named strings. The form of the decaration is:

> <!ENTITY *name* "*string*">

is that part of the document which follows the declaration the sequence of characters &*name*; will be replaced (when interpreted) by the *string* in the declaration. IDC uses this device with one restriction. The declarations must be occur immediately before a valid tag. Also, and this is true of SGML, for a entity to be used in the value of an attribute, the attribute value must be enclosed in quotation marks. Entities can also be used in resource indexes where they are particularly useful.

There are three entities which are automatically decared these are:

```
<!ENTITY amp "&">
<!ENTITY gt "<">
<!ENTITY lt ">">
```

which allow the three symbols &, < and > to be used in a marked up document.

The Sampler Formats

This section consists of annotated copy of the source document (SAMPLER.IDC) for the example interactive document (SAMPLER.TBK). The text of the document is in plain text and the annotations in italics. The annotations refere to the part of the document which precedes the annotation and elements of the document which are underlined are specifically mentioned in the annotation. Annotations in square brackets give information which can be ignored, but which may be of interest to those wishing a greater insight into the system.

```
<!-- IDC Sampler 1.0 17th December 1993  -->
```

A comment.

```
<header start=Menu>
<title>An IDC Sampler</title>
<copyright>1993 Computer Based Learning Unit, The University of Leeds</copyright>
<caption>IDC Sampler</caption>
<credits>Designed and produced by Kenneth Tait</credits>
<date>8th December 1993</date>
</header>
```

The copyright symbol © is prefixed to the text in the display of the content of <copyright>.

```
<body>

<f1 name=Menu>
Illustrative Interactive Documents
<o go=MCQ>Multiple-choice Questions</o>
<o go=Test-Teach>Test-Teach Exercises</o>
```

```
<o go=Presentation>Presentation</o>
<o go=Hyperbook>Hyperbook</o>
<o go=Linking>Linking</o>
</f1>
```

> *The content of <f1> should be compiled with the Menu format which means that before compiling the document an association must be made between f1 and the format* **Menu** using the Options page of the compiler. *(The fact that this page is called "Menu" is coincidental.)*
>
> *The first part of the content is text that describes the menu. Each option in the menu is marked up with the <o> tag and the destination page when the option is selected is specified through the* ***go*** *attribute.*

```
<f2 name=MCQ next=MCQ1>
<h1>Multiple-choice Questions</h1>
The following sequence of pages illustrates some possible styles of multiple-choice question.
<p>As with all formats changes in colour, layout of the screen and details of the interaction could be
changed to meet particular requirements.
</f2>
```

> *The content of <f2> should be compiled with the Hyperbook format. This page does not use many features of the format.*
>
> *The attribute* ***next*** *gives the name of the page which will be displayed when the user clicks the NEXT button. If a page has no* ***next*** *attribute then the NEXT button will not be visible. Because pages are linked together in ways like this the actual order of pages in the source document is irrelevant.*
>
> *The content of tag <h1> is considered to be a header and will be distinguished from the rest of the text according to the options set when using the compiler. The digit 1 signifies that this is the highest level heading. A maximum of five levels is catered for. All headings are noted by IDC and used to construct a active contents list which can be viewed by the user.*
>
> *Because line breaks have no significance the tag <p> is used to indicate the start of a paragraph. This is one of the few tags which does not require a matching closing tag. The content is taken to be everything up to the next <p> tag or the next tag which cannot be part of the text.*

```
<f2 name=MCQ1 next="Simple MCQ">
<h2>A simple multiple-choice question</h2>
In this question the learner has only <emp>one</emp> chance to pick the correct answer. This results in
either confirmation of the answer or the correct answer being displayed. In either case a short
explanation is given.
<p>There is an active phrase in the explanation which when clicked gives further information. This active
phrase could have been a link to another page, another document or another application.
<p>Notice that the menu bar is not displayed for these pages.
</f2>
```

> *The content of <f2> should be compiled with the Hyperbook format.*
>
> *The value assigned to the attribute* ***next*** *is enclosed in quotation marks because it contains a space.*
>
> *The tag <h2> is used to mark a second-level header.*
>
> *The <emp> tag marks text which is to be emphasised. The way this is done is determined by the options set in the compiler.*

```
<f3 name="Simple MCQ" image=graph next=MCQ2>
In any interval of time which of the following is represented by the total area between the curve and the
time axis?
<o ans=y>distance travelled during the interval</o>
<o>distance from starting point</o>
<o>mean acceleration during the interval</o>
<o>average speed over the interval</o>
<o>maximum velocity reached during the interval</o>
<x>If velocity is in metres per second and time in seconds then a unit of area which is one second by 1
metre per second will represent <a go=area>one metre, that is distance</a>.</x>
</f3>
```

> *The content of <f3> should be compiled with the MCQ format.*

*The attribute **image** gives the name of the image which should be displayed. The image must be in the bit-map format and stored in a file by itself. A link between the name given here (**graph**) and the file containing the image is made in the Resource Index. On compilation the image becomes part of the interactive document and does not need to be available at run-time.*

*Each option is marked up with the <o> tag. The single option which is the correct answer should have the attribute **ans** with the value **y**. There can be as many options as desired, but devising a large number of good distractors is difficult and the presentation may become unsatisfactory.*

*[It might have been better in creating this format to have used two tags rather than just <o>: for example, <a> could have been used for the correct answer and <d> for the distractors. This would have avoided the need for the attribute **ans**. To change to this markup would require a small change to the format.]*

*The tag <x> is used to mark up the explanation. The explanation can be marked up as hypertext, that is it uses the same markup as the text in the Hyperbook format. In this page active text is marked using the <a> tag with the **go** attribute giving the name of the destination page. Although it is not apparent from the markup the destination page in this case (**area)** is a remark and will appear in a pop-up window when the active text is clicked. The explanation is optional.*

```
<fn name=area>One metre per second multipled by one second gives one metre: so area represents
distance.  If the interval of time was zero then the area would be zero so the area must represent
distance travelled in the interval.
</fn>
```

*This is a remark. It can be used by any format which incorporates a pop-up, that is has a window which is normally hidden and in which text can be displayed. Both the MCQ and the Hyperbook formats have pop-ups. This remark is used by the page **Simple MCQ**.*

```
<f2 name=MCQ2 next=Pairs>
<h2>Matched Pairs</h2>
The next example requires the learner to match items from one list with another.  In this case there are
exactly the same number of items in each list.  A variation of this format would allow extra items in the
"answer" list.
<p>This particular example is based on an Italian exercise used with a system called LUISA developed
at the University of Leeds by Brian Richardson and his colleagues.
</f2>
```

Another page with content that should be compiled with the Hyperbook format.

```
<f4 name=Pairs next=MCQ3>
Combina la traduzione inglese con quella italiana
<i>the tail<a>la coda</a></i>
<i>the fur<a>il pelo</a></i>
<i>the lair<a>la tana</a></i>
<i>the trunk<a>la proboscide</a></i>
<i>the tusk<a>la zanna</a></i>
<i>the whisker<a>il baffo</a></i>
<i>the paw<a>la zampa</a></i>
<x>Notice the links with some English words: "pelt"  meaning the skin of an animal; "proboscis" a fancy
word for trunk or similar appendage; and "coda" is a musical term (from the Italian) for a short piece at
the end of a longer work.</x>
</f4>
```

The content of <f4> should be compiled with the Classification format.

The content begins with an instruction and then a double list marked up with <i> and <a>. The first part of the content of <i> gives the item that has to be matched by the item marked up with <a>. Notice that the whole of the <a> tag is part of the content of <i>.

The explanation or notes are marked up with <x>.

```
<f2 name=MCQ3 next="Stem and Branch">
<h2>Stem-and-branch Question</h2>
This style of MCQ is widely used in the biological sciences and medicine.  The example you will see is
taken from a large bank of such questions in Pharmacology which has been used at Leeds in various
forms for more than ten years.
<p>A section of the bank is supplied with this IDC Sampler as a separate system.
</f2>
```

Another page with content that should be compiled with the Hyperbook format.

```
<f6 name="Stem and Branch" si=amphetamine next=Menu>
Possible biotransformation pathways for amphetamine include
<br ans=t>hydroxylation
</br>
<br ans=f>dehalogenation
</br>
<br ans=t>dealkylation
</br>
<br ans=f>deamination
</br>
<br ans=f>methylation
</br>
<x>MAO. No catechol groups thus COMT inactive. Look at functional groups.
Phenyl ring hydroxylated. Nitrogen is dealkylated.
<p>Note that deamination is very slow because of the alkyl group on the
alpha carbon of the sidechain.</x>
</f6>
```

The content of <f6> should be compiled with the MCQ:S&B(I)Y format. This format is a variation of the MCQ:S&B format used in the Pharmacology MCQ Bank.

*The attribute **si** specifies the image that should be should with the stem. [If an image is to be shown with the explanation then the attribute **xi** can be used. If this latter image is large then some of the explanatory text will be obscured. The format could easily be modified to give the user control over the display of explanation image.*

[The explanation text can be marked up as Hyperbook text put there is no pop-up so any active text must be linked to other pages, documents or applications.]

```
<f2 name=Test-Teach next=Scotland>
<h1>Test-Teach</h1>
These are question-and-answer exercises which assume that the learner already has some knowledge,
but is likely to make mistakes when answering questions.
<p>Exercises of this kind should be sufficiently difficult that some learners make mistakes. However, it is
important that mistakes are recognised so that informational feedback can be given to help the learner
recognise and overcome the error.
<p>First a simple exercise as an illustration...
</f2>
```

Another page with content that should be compiled with the Hyperbook format.

```
<f7 name=Scotland next=Again>
Which is the largest city in Scotland?
<cr>Glasgow</cr>
<ar allow=s>Glasgow<fb>I think you mean have the right city, but you failed to spell it
correctly.</fb></ar>
<ar allow=sx>Glasgow<fb>I think you mentioned the the right city - just type it's name by itself.</fb></ar>
<ar allow=x>Edinburgh<fb>Edinburgh is the capital of Scotland, but not its largest city.</fb></ar>
<ar allow=sx>Edinburgh<fb>I think you mean Edinburgh. Edinburgh is the capital of Scotland, but not its
largest city.</fb></ar>
<ar allow=sx>Aberdeen<fb>Although the centre of Scotland's oil industry it is not the largest
city.</fb></ar>
<ca>The largest city in Scotland (in terms of population) is Glasgow.</ca>
<ht>The largest city in Scotland stands on the River Clyde and was once the centre of a large ship-
building industry.</ht>
</f7>
```

The content of <f6> should be compiled with the Test Teach format.

The text of the content is a question. The remainder of the content gives correct and anticipated responses together with feedback, the text of a completely correct answer possibly elaborated and a hint. We will take each tag in turn.

*The tag <cr> marks the correct answer as it must be typed by the user. Variation from this can be allowed using the attribute **allow** in the way described for <ar>.*

The tag <ar> marks and anticipated response as it is expected to be typed by the user. In addition *it contains feedback marked with <fb> which will be displayed if the user's response matches the anticipated response. The matching is controlled by the attribute* ***allow*** *which takes a value made up of one or more of the letters* ***s****,* ***x****, and* ***o****. If* ***s*** *is present then a simple algorithm is used to allow spelling variations. If* ***x*** *is present then extra words not mentioned in the anticpated response will be ignored. If* ***o*** *is present then the order in which the words of the response are typed need not be that of the anticipated response. It should be noted that the matching is not case-sensitive. Furthermore, if the value of* ***allow*** *is* ***n*** *and the anticipated response is two numbers, say, n and e, then the matching will accept any number in the range n-e to n+e.*

The tag <ca> marks the text which should be given to the user as the correct answer.

The tag <ht> marks the text which should be displayed as a hint.

```
<f2 name=Again next=Menu>
The Test-Teach format incorporates some simple matching algorithms which can tolerate different word order, allow extra words to be ignored and recognise an approximate spelling.
<p>The algorithms are not sophisticated, but could be improved if necessary.
<p>Although it cannot be tested with this example, the format will accept numerical answers within a specified range.
<p>If you would like to try the question again, click BACK...
</f2>
```

Another page with content that should be compiled with the Hyperbook format.

```
<f2 name=Presentation next=Introduction>
<h1>Presentation</h1>
What follows is a short presentation using a simple format. Such a presentation could be used in front of an audience with the screen image projected onto a large display.
</f2>
```

Another page with content that should be compiled with the Hyperbook format.

```
<f5 name=Introduction next=Leeds>
Getting Started with CBL
<pt>Off-the-shelf software</pt>
There is an abundance of CBL materials. The CTI Centres for particular disciplines maintain catalogues of these materials. Will it adapt to you, or will you (and your students) have to adapt to it?
<pt>Write it yourself</pt>
Authoring of CBL materials is a complex task. It requires understanding of the subject matter, pedagogical insight, design expertise, an appreciation of the strengths and weakness of the technology and the ability to think like a programmer. This remains true even when using a powerful and sophisticated authoring system.
<pt>Buy in a programmer</pt>
May well get the materials produced, but will you be able to maintain them and develop them as both educational requirements change and technology improves?
<pt>Collaborative development</pt>
Expertise and experience already exists. It exists in this university. Talk to people who have learned important lessons.
<pt>Departmental project</pt>
In the end you will only be successful in using CBL materials if CBL is an integral part of your department's plan.
</f5>
```

The content of <f5> should be compiled with the Points format.

The content of <f5> is text broken up by several occurrences of the tag <pt>. The text before the first <pt> can be considered as a title or introduction. Each occurrence of <pt> gives a point (or heading) and the text which follows it is an elaboration of that point.

There can be a maximum of five points.

```
<f5 name=Leeds next=Developments>
What is happening at Leeds?
<pt>Biochemistry</pt>
The Biological Sciences require all their students to be familiar with word processing and the use of spreadsheets. The CBL materials used in Biochemistry include simulations and diagnostic case studies. Most of this has been developed at Leeds.
<pt>Languages</pt>
Both Russian and Italian have extensive software developed here at Leeds to support language learning such as vocabulary and grammar exercises.
```

```
<pt>Pharmacology</pt>
For over ten years a bank of about 1500 interactive multiple-choice questions (MCQs) with explanations
has been made available to students of Dentistry and Medicine as well as those on Pharmacology
courses.  To this has been added material from other sources as well as exercises which make use of
spreadsheets.
<pt>Microbiology</pt>
A small bank of MCQs has been developed using the same system as Pharmacology.  In addition there
are a number of exercises which use the commercial Question Mark system.
<pt>Other Departments</pt>
Physical Chemistry, English, Computer Studies, Engineering all make use of some CBL materials and
there are many other departments that expect their students to use standard
software.
</f5>
```

Another page with content that should be compiled with the Points format.

```
<f5 name=Developments next=Authoring>
What's happening in CBL?
<pt>TLTP</pt>
The Teaching and Learning Technology Programme is a nation-wide intitiative (1992-1995) aimed at
producing and supporting the use of CBL materials in higher education.  A number of departments in the
University are involved in TLTP projects.
<pt>CBL and Multimedia Support Unit</pt>
This is TLTP-funded and provides advice and supports a range of projects in the University.
<pt>The Computer Based Learning Unit</pt>
This unit was established at Leeds almost 25 years ago to investigate the potential of CBL.  It's activities
cover both research and developement.  Current research is concerned with advanced knowledge-based
learning systems.  Development work is carried out in collaboration with external bodies or departments
within the University.
</f5>
```

Another page with content that should be compiled with the Points format.

```
<f5 name=Authoring next=Menu>
Design and Production of CBL
<pt>Authoring</pt>
This is a multi-skilled activity.
<pt>A different approach</pt>
This separates out aspects of the design and automates the production.  This is approach is suitable for
the mass production of commonly used styles.
<pt>Aspects of design</pt>
Structuring of content. Design of the presentation. Programming of the interaction. Specification of the
control.
<pt>Interactive Document Compiler</pt>
This converts marked-up text together with an index of resources (such as images and other
applications) to create an 'interactive document'.  This can be a hypertext document or a bank of MCQs
or any kind of interactive activity which uses structured content.
<pt>Formats</pt>
IDC uses the concept of a 'format' which combines presentation and interaction.  There are several
versions of IDC each with a set of formats.  Each format requires the text input to be marked-up
appropriately, but a single interactive document may use several formats.
</f5>
```

Another page with content that should be compiled with the Points format.

```
<f2 name=Hyperbook next="University of Leeds">
<h1>The Hyperbook Format</h1>
This page has been produced using the Hyperbook Format.
<p>As you will see in addition to displaying text there can be a <a go=colours>picture</a>.
<p>Also links to other parts of a document (or to other documents and applications) can be added.
These links may be followed by clicking on one of the buttons that may appear at the bottom right or
through active text like <a go=OK>this</a>.
<p>When you click the NEXT button you will see a more extended example.</f2>
```

Another page with content that should be compiled with the Hyperbook format. This has two active elements. The first is linked to another page; the second will activate a pop-up.

```
<fn name=OK>Clicking active text or a button may bring up a window like this or take you to another
page, or to another document.
</fn>
```

Another remark. Remarks can be used by any format with the fucntionality to handle them. The same remark can be used by several pages.

```
<f2 name=Colours image=C16>
<h2>Images</h2>
Most of the images in this sampler have only 16 colours, like the picture shown here of the Great Hall at
the University of Leeds. Quality would be improved if images with more colours had been used, but as
yet not all systems are equipped to display a greater number of colours. For comparison a <a go="More
Colours">256 colour version</a> of the same picture is available.
</f2>
```

Another page with content that should be compiled with the Hyperbook format.

*The attribute **image** gives the name of the image which should be displayed.*

```
<f2 name="More Colours" image=C256>
<h2>The Great Hall, The University of Leeds</h2>
If this picture looks no better (or worse) than the one on the previous page then either your monitor is not
capable of displaying an image with 256 colours or the display driver currently in use will only handle 16
colours.
</f2>
```

Another page with content that should be compiled with the Hyperbook format.

```
<f2 name="University of Leeds">
<h1>The University of Leeds</h1>
This example contains information about the <a go=University>University of Leeds</a>, about the <a
go=City>City of Leeds</a> and something about the <a go=Region>region</a> in which it is set.
</f2>
```

Another page with content that should be compiled with the Hyperbook format.

```
<f2 name=University image="Great Hall">
<h2>The University of Leeds</h2>
Leeds is one of the largest and most successful universities in the country with a long-established
international reputation for the high standard of its teaching and research. The University received its
charter in 1904, but its <a go=Origins>origins</a> lie in the nineteenth century.
</f2>
```

Another page with content that should be compiled with the Hyperbook format.

```
<fn name=origins>The University has grown out of the
Yorkshire technical College and the Leeds Medical School,
both of which were founded in the nineteenth century.
</fn>
```

Another remark.

```
<f2 name=City image="Town Hall" more=Green>
<h2>The City of Leeds</h2>
Leeds is a prosperous, commercial, industrial and manufacturing city, and is also the cultural and
sporting centre for much of the region. Impressive new shopping precincts complement traditional
Victorian arcades and have created a fine regional shopping centre.
</f2>
```

Another page with content that should be compiled with the Hyperbook format.

*The attribute **more** gives the name of the page which will be displayed when the user clicks the MORE button. If a page has no **more** attribute then the MORE button will not be visible.*

*[There is an **also** attribute and ALSO button which can be used in the same way.]*

```
<f2 name=Green image="Park">
<h2>A  green city</h2>
Leeds is a 'green' city, proud of its parks and open spaces including Woodhouse Moor, close to the
University, Roundhay Park and the Temple Newsam estate.
</f2>
```

Another page with content that should be compiled with the Hyperbook format.

```
<f2 name=Region image=map next=Country>
<h2>Where is it?</h2>
Leeds lies in the heart of West Yorkshire close to the geographical centre of Britain.  It has good
motorway and rail connections to the rest of the country.
</f2>
```

Another page with content that should be compiled with the Hyperbook format.

```
<f2 name=Country image=Country>
<h2>Areas of natural beauty</h2>
Within easy reach of Leeds are many areas of outstanding natural beauty: the <a
go=Wharfedale>Yorkshire
Dales</a>, the North York Moors and the Vale of York.
</f2>
```

Another page with content that should be compiled with the Hyperbook format.

```
<f2 name=Wharfedale image=Dales>
<h2>A view of Wharfedale from Beamsley Beacon</h2>
</f2>
```

Another page with content that should be compiled with the Hyperbook format.

This page is used to give a caption to a picture.

```
<f2 name=Linking next="Link IDC">
<h1>Linking to other documents and applications</h1>
It is possible for a button or active text to either <a go=asynchronous>launch</a>  another Windows
application or to enter another document that has been produced using IDC.
<p>In this latter case Quitting the second document will cause a return to the page of the first document
from which the second document was called.  This enables several documents to be linked together in a
<a go="MCQ Bank">unified system</a>.
</f2>
```

Another page with content that should be compiled with the Hyperbook format.

```
<fn name=asynchronous>
Windows applications are launched ASYNCHRONOUSLY.  This means that the document from which
the new application was launched retains no control and the document and the
application co-exist independently.
<p>The user remains responsible for closing the application.
</fn>
```

Another remark.

The only tag which can be used in a remark is <p>.

```
<fn name="MCQ Bank">
The Pharmacology MCQ Bank which in its entirety consists of about fifty documents is an example of
such a system.  A fragment of the bank is supplied with this sampler.
</fn>
```

Another remark

```
<f2 name="Link IDC" more="IDC example" next=Windows>
<h2>Linking a document produced with IDC</h2>
Both the MORE button and the active text will call up a completely separate document created with IDC.
<p>Click <a go="IDC example">this</a> or the MORE button to go to the document.
<p>To return to this page you must QUIT the document.
</f2>
```

Another page with content that should be compiled with the Hyperbook format.

*The name **IDC Example** is associated with the appropriate document through the Resource Index for this document -- Resource Indexes are described in the section on Running the Compiler. The Resource Index for this document will be found in the file RESOURCE.IDX. When an interactive document is opened from another interactive document then the title page of the second documentis not displayed.*

```
<f2 name=Windows next=Menu>
<h2>Launching Windows applications</h2>
Any Windows application can be launched from an interactive document.  These are only a small number
of examples.  The active text could be anything and the launch could be triggered by a button.
<p><a go=Notepad>Launch NOTEPAD</a> [The file opened will be the README.TXT supplied with this
sampler.]
<p><a go=WinHelp>Display the Windows Help File</a>.
<p><a go=Toolbook>Another Toolbook document</a> using a separate instance of Toolbook.
</f2>
```

Another page with content that should be compiled with the Hyperbook format.

*The names **Notepad**, **WinHelp**, and **Toolbook** are associated with the appropriate commands through the Resource Index for this document.*

```
</body>
```

The end of the document. Nothing after the closing tag </body> is of significance.

Running the Compiler

The Interactive Document Compiler which forms part of this sampler is a Toolbook book which can only be run with the full Toolbook authoring system, and as it temporarily modifies itself the two files comprising the compiler should be marked as 'read-only' . The two files are:

IDC_136.TBK
SYS136.TBK

and both must be in the same directory.

When the compiler is invoked by opening its icon an introductory screen is displayed with a panel giving the version of the compiler, some information which identifies the particular compiler, the date at which the shell was created and a list of the formats which the compiler can handle. After a short delay the control screen appears. This screen has five control buttons (ignore the smaller buttons at the bottom for the moment) and brief instructions which are amplified in this document. The right hand side of the screen is given over to a window where messages identifying errors in the source document are displayed.

[You may fnd it useful before reading the rest of this section to try out some of the suggestions in the section SOME THINGS TO TRY.]

An interactive document as specified by an source document may require further resources to be linked to it or compiled into it. These resources may be images stored in separate files, other interactive documents which will be invoked by the document being compiled, or other applications. All the resources for a particular interactive document must be listed in a special document called the resource index though several source files may share the same resource index. The resource index has the following structure:

```
<index>
index-entry
.
.
.
</index>
```

where index-entry can be either:

```
<idc name=uniquename file=filename>description</idc>
```

or

```
<image name=uniquename file=filename>description</image>
```

or

<app name=*uniquename* file=*filename* arg=*argument*>*description*</app>

where *filename* is, in the first case, the full path and filename of another interactive document created using IDC (with .TBK as filetype), in the second, the full path and filename of a file containing an image in bit-map format usually with .BMP as filetype and in the third an executable file. In this last case the attribute **arg** is optional and can be used to give a command-line argument to the application . Not only must each *uniquename* be unique within the index, neither must they clash with names used for pages or remarks in the interactive document being compiled. The description in each entry is at present ignored. It can be used by the author or designer to document the resources index. Comments may be included and entities declared and used in a resourceindex in the same way they are in a source document.

Options

Options affect the way a compiler works. Once the options are set then each compiler keeps a record of those options in a file of the same name as the compiler but with the OPT as the file type. This file is always stored in the root directory of the default disk.

The General Options screen appears when the user has clicked the Options button on the Control screen. The General Options allow the user to associate formats with the tags f0, f1,..., f9 which delimit pages in the source document, to specify the name of the file containing the resource index and to give the default filename of the source document. It is also possible (using the **Copy Options to file** and **Load Options from file** buttons) to save options in a file and to load options previously saved as a file. The file SAMPLER.OPT contains the options required to successfully compile the document SAMPLER.IDC. When specifying files in the dialogue box it is necessary to give the full path and name of the file.

Clicking on the button **Set Resource Index** button causes a dialogue box to appear. The user should type the full path and name of the file containing the appropriate resources index. If IDC already knows about a resource index then that name will appear in the box and can either be edited or deleted. The filetype of an index file is normally IDX. The name of the current resource index is displayed alongside the button used to set it. In a similar way the user can use the **Set Default File** button to specify the filename of the source document, or the other buttons to store and load sets of options. It is possible to direct the compiler to store the compiled document in a different directory by setting the destination path.

The default file is the one which will be offered when compiling is initiated. (At that stage the user may decide to compile a different source document, but often the user works with the same file for some time and it more convenient if the name of that file is set as the default file.) Input files for IDC normally have IDC as their filetype. The name of the current default file is displayed alongside the button used to set it.

The **Text Options** button displays the Text Options page. Text options apply to marked up text. Only certain formats allow such text. The text options apply to all such formats in a dcoument. It is not posible to vary the options within an single document. It is not necessary to set any text options in which case the default text options will be used. All the options are selected by clicking on the square to the left of the desired option. Where options are mutually exclusive then choosing one will cancel the others.

Anchors are what distinguishes hypertext from plain text. Anchors are active text which is linked to another page or document or application. They can be identified in different ways. The user selects the desired one before compiling.

Boxed This is the default for Toolbook. Each anchor is enclosed in a box.

Inverted In this case each anchor is shown in a pair of colours (one for the text, one for the background) which are complementary to the colours used for the rest of the text.

Bold, italic and underline distinguish anchors from the rest of the text by adding the chosen style to the font of the text.

If one of **bold**, **italic** or **underline** is selected then the anchors will be distinguished all the time. If either **boxed** or **inverted** is chosen then the reader will be able to choose whether the anchors are distinguished or not by selecting the Show Links option from the Actions menu available at run-time.

The default for anchors is **boxed**.

Text which is tagged for emphasis (using <emp> and </emp>) can be indicated using any one of **bold**, **italic** or **underline** combined with characters in **Upper case**, **Title case** (initial letter of each word capitalized), **Sentence case** (first letter capitalized) or **Unchanged** .The default is **italic**, **Unchanged** text.

There are five levels of heading with the corresponding tags <h1>, <h2>, <h3>, <h4> and <h5>. Each one can be displayed in a style of text chosen from **bold**, **italic** and **underline** with characters in **Upper case**, **Title case**, **Sentence case** or **Unchanged**. The user can also determine how a particular heading level is set apart from the rest of the text. The heading can be preceded by a **Blank line** or just start on a **New line**; and the text following it can be separated from it by a **Blank line**, or start on a **New line** or follow the heading on the same line with a **Space** separating it from the heading.

The defaults are:

h1	Bold, Upper case, Blank before, Blank after
h2	Bold, Title case, Blank before, New line after
h3	Bold, Sentence case, Blank before, Space after
h4	Underline, Sentence case, Blank before, Space after
h5	Underline, Sentence case, New line before, Space after

Paragraphs are indicated using the tag <p> and can be marked on the screen by having a **Blank line** preceding the paragraph, starting the paragraph on a **New line** or starting it on a new line and **Indenting** it.

Each item in a list can be prefixed by a 'bullet'. The user can chose one of the symbols shown, or click on the ? and type into the box alongside the four-figure code for the desired character . The tag for a list is <l1> with items in the list marked with <item>. Further work needs to be done in respect of lists.

The user returns from the Text Options screen to the General Options screen by clicking on the OK button, and similarly to return to the Control screen from the General Options screen. This last action causes the options to be saved in a file with the same name as the compiler but with the file type OPT. This file is always stored in the root directory of the default disk. When the particular IDC is used used again it will load the the options from option file if one exists. If the file cannot be found then the options will be set to the defaults. If the user wishes to save a set of options then they can be copied to a file using the **Copy Options to file** button on the General Options screen. This makes a copy of both General and Text Options which can be restored by using the **Load Options from file** button. In whatever way the options have been set it is the current ones that are saved to compiler's own option file whenever the user returns to the Control Screen; those in any other file can only be changed by using the the **Copy Options to file** button.

Compiling

Compiling is initiated by clicking the Compile button. This will cause a dialogue window to appear in which the name of the default file is already visible and highlighted. This can be deleted or edited in the usual way, or accepted by immediately clicking the OK button. Compiling has two phases. During the first phase, the resource index is located and read, and then source documentis read and a list of all names encountered made. Some preliminary checks are made on the syntax of the input file, but these cannot identify all errors. While the file is being scanned in this way the cursor will alternate between a large and a small black square. If no errors are detected at this phase then the second phase which creates the pages of the new book begins. At the beginning of the second phase 'Interactive Document Compiler' will disappear from the title bar and be replaced by the caption of the document (if it has one) and the cursor will begin to alternate between an empty square and a black square. (The user may initiate the first phase only by clicking the Read button. If this is successful then Compile can be clicked to perform both phases.)

If errors are encountered during the first phase then the second phase will not be started and the number of reported errors will appear above the right hand window which will contain a a trace of the process carried out including notices of the errors detected. This window may have to be scrolled to view all the information. Much of the information may not be useful, but error messages will stand out and should help the author modify the source document (or possible the options) as required.

If the error is in the input file then the user may initiate another task and edit and save the input file and the return to IDC and try compiling again. This should only be attempted if the title bar still contains 'Interactive Document Compiler'. If it does not then the user must quit (using the Quit button) and re-start the compiler before attempting to compile the modified document.

If errors are detected at this stage (after the caption in the title bar has changed) then the information about them will appear in the trace. These can be corrected as described earlier, but as new pages will already have been added to the IDC then the user should quit the IDC (by clicking on the Quit button) and re-start the IDC before attempting to compile again.

If no errors are detected in this second phase of compiling then a trace will appear in the right hand window and the message 0 Error(s) will be displayed above it. This means that the user can move on to Detaching.

Detaching is a short process during which pages of the current book which belong to IDC are deleted and the remaining pages saved as a new book. Detaching is initiated by clicking on the Detach button which will beep if errors have been detected during compiling. Before detaching the new book the user will be prompted for its name. The name offered will be the same name as the input file with the file type changed to TBK . This suggested filename may be edited if necessary, but it is simpler and more convenient to leave the filetype as TBK. IDC will replace any existing book of the same name without warning the user.

During detaching the cursor alternates between a small and a large black circle and on sucessful completion Toolbook quits. Just before it does so the first page of the new book will appear momentarily.

> *Note that if there is a book of the same name and it is read-only then Toolbook gives an inaccurate error message informing you that the compiler (for example*

> *IDC_136.TBK) is read-only and cannot be replaced. If this occurs then either rename the offending book, detach as a different name, or change the attribute so that the file is not read-only and can be replaced.*

It should be noted that Version 1.36 is essentially a beta release. In particular the error recognition may be deficient sometimes resulting in a Toolbook error rather than an IDC error. Should a Toolbook error occur then click CANCEL in the dialogue box, use control-X to exit from Toolbook and if prompted to save changes then select NO. (Saving changes to the compiler is password protected so even if YES is chosen the action cannot be completed without knowledge of the password.)

Customizing the Compiler

The Control Page contains a number of small buttons which have not been mentioned. These are:

ATTACH FORMATS
MINIMIZE
ABOUT IDC
EXTRACT FORMATS
CREATE SHELL
and
NAME

ABOUT IDC displays information as to the origin of IDC and contact information.

EXTRACT FORMATS creates files in which copies of formats attached to the compiler are stored independently. Any number of the formats currently attached to the compiler can be copied into a single file. When asked for a list the user may give the name of a single format, reply with * to indicate all formats or type a comma-separated list of formats. The path and name of the file in which the formats are to be stored will be requested.

NAME changes the string which is displayed (in red) on the title page. This will have no permanent effect unless a new version of the compiler is created with one of ATTACH FORMATS, MINIMIZE or CREATE SHELL.

The remaining buttons create new versions of the compiler. In all these cases it is not possible (without knowledge of a built-in password) to create a new version of the compiler with the same filename as the current one. Of course it is possible to create it with a temporary name and then rename the compiler using the File Manager for example. Although irritating this does prevent accidental loss of formats. However, provided the filename given is not the same as that of the current compiler, the user will not be warned if there is already a file of the given name and that file will be replaced by the new version of the compiler. In giving the new filename the user should provide the full path and name.

ATTACH FORMATS will attach the formats from a file created by EXTRACT FORMATS. The process is a little awkward and is fully described in Using a Different Format for the Menu in the section SOME THINGS TO TRY.

MINIMIZE provides a convenient way of removing (unwanted) formats from a compiler. The effect of the action is to delete from the compiler any format which is not currently associated with one of the tags f0,f1,...,f9. Thus it is only necessary to go to the Options page and to make sure that the formats to be retained (and no others) are associated in that way and then return and click MINIMIZE. Note: clicking one of the buttons f0,f1,...,f9 with the *right* mouse button will clear any association.

CREATE SHELL creates a copy of the compiler with no formats at all. This can be used to add formats to to create a new compiler. The name of the compiler (in red on the title page) will be set to 'Shell'

<u>DDE</u>

An IDC compiler will respond to DDE requests. For example it will respond to the request (from another instance of Toolbook) of the form:

> setRemote *optionsfile* to *sourcefile* application Toolbook *idccompiler*

where *sourcefile* is the file containing a source document, *optionsfile* is a file containing the appropriate options (as saved with the **Copy Options to file** button), and *idccompiler* is the name of the actual IDC compiler to be used. This mechanism can be used to create a Toolbook which will (for example) automatically compile a set of documents.

Using an Interactive Document

All interactive documents created by IDC require the run-time version of Toolbook and to make an interactive document available through Windows it is only necessary to create an icon with a command line of the form:

> TBOOK *filename-of-interactive-document*

provided that TBOOK.EXE and its components are in a directory that is in mentioned in the PATH variable. The document can be used by double-clicking the icon. Alternatively if an association has been made between .TBK and TBOOK.EXE double-clicking the name of the document in the File Manager will open it.

An interactive document CANNOT be opened successfully unless a copy of COMM136.TBK resides in the same directory as the interactive document.

An icon file is provided with the sampler called IDC136.ICO which may be used to identify interactive documents.

Where the interactive documents are to be used over a network then TBKNET.EXE should be available and all interactive documents which will be used by more than one user simultaneously should be set to read-only as stated in the Toolbook documentation. In particular COMM136.TBK should be read-only if it is on a network drive even if it will only be used by a single user as it may be used several times when linked intereactive documents are accessed.

<u>Menus</u>

All interactive documents created with IDC have the same menus. Some formats may hide the menu bar preventing access to the operations which are only available through the menus unless specific alternative means are provided by the format through buttons, for example.

There are three menus: Actions, Pages, and Bookmarks. Not all options on all menus are always available and a particular document can be opened in different ways which may vary the options which are barred. When an option is not permitted either it will not be present (either the menu will be absent or the option will be missing from the menu) or the option will be displayed in grey.

The **Actions** menu has three, four or five of the options **Help**, **Show Links**, **Notes**, **Copy Text** and **Quit**. (Neither **Notes** nor **Copy Text** are used by the sample formats.)

Show Links is only present if the document has be compiled with the anchor option as **boxed**, or **inverted**. When anchors are shown in neither of these styles then they are permanently shown. Otherwise the **Show Links** option can be used to distinguish anchors from other text or to remove the distinguishing characteristic. Anchors whether marked or not are always sensitive and the mouse pointer will change to a white open rectangle when ever it passes over one.

The header of an interactive document may specify one or more Help Documents using the <help> tag. If this is the case then there will be a **Help** option in the **Actions** menu. Selecting this will bring up a menu of help documents from which the reader can choose. If there is only one such help document then the menu will not be displayed and the help document will be invoked immediately. A help document can be either another IDC, and application or part of the current document and is just like any other destination. (The source documents for the Pharmacology MCQ Bank will serve as illustrations for the use of the <help> tag.)

Quit exits from the current interactive document. If the interactive document was invoked by another interactive document then control is returned to the page of that document from which the invocation was made. Otherwise, the document will be closed and the current instance of Toolbook terminated. **Quit** may also cause information to be stored in files for future use such as the one which keeps informationabout bookmarks.

The **Pages** menu has two options: **Contents** and **Visited**.

Selecting **Contents** displays a scrollable menu of headings taken from pages which contain marked up text. (In principle it is possible for any format to add an entry to the headings.) Clicking once on any heading takes the user to the page on which that heading appears. Headings are displayed in the same style as in the text, with ‘h2’ headings indented from ‘h1’ and so on .

Selecting **Visited** displays a list of the names of all the pages that have been visited in the current session with a document. Each page only appears once even if it has been visited several times. If there are more names than will fit the window a scroll bar will be provided. Clicking on any name in the list will cause the page with that name to be displayed . The name of the current page will be displayed on the screen, but each format may use a different location. The order of the pages is such that those which which were more recently visited for the first time are nearer the top of the list. This means that the bottom item in the list is the opening page of the document.

The **Bookmarks** menu allows pages to be ‘marked’ permanently, by which it is meant that the ‘marking’ is retained between different occasions on which the document is used. This means that having left the title page of a document it is possible to go immediately to one of a number of pages that were found to be of significance on previous occasions. Marking pages can be used to make navigation around a document more convenient. Pages often display their names. When this is so the name of a marked page will have (*) appended to its name.

This **Bookmarks** menu has four options : **List marked pages**, **Mark this page**, **Unmark this page**, and **Remove all**.

List marked pages displays a list of marked pages. When the user clicks on any one of the marked pages that page will be displayed.

Mark this page causes the current page to be marked.

Unmark this page causes the mark to be removed from current page.

Remove all removes all the bookmarks in the current document.

Information Files

Each interactive document may have an information file. This is a plain text file with the same name as the interactive document, but with the filetype INF and to be effective it must be in the same directory as the interactive document itself. An information file allows parameters and options associated with a document to be set. For example the directory in which data files are to be stored (which is the root directory of the current disk by default) can be specified. The format of the information file is a set of lines. If a line contains an equals sign (=) then that is interpreted as an instruction to set a value for a parameter or option. For example:

```
datapath=C:\RECORDS
```

would indicate that all datafiles had to be stored in the directory C:\RECORDS . Lines which do not contain an equals sign are ignored, except that the line consisting of the single word end (which must be present) is taken to be the end of the file. The case of text in information files is not significant and if there is no information file then all relevant parameters and options are set to their defaults. Omission of an assignment for a parameter or option also means that its default value will be used .When one interactive document opens another then the information file for the second document is not consulted. Instead the values for the parameters and options in force in the first document are used with the second. This means that all the information required for a set of interactive documents which call each other must be in the information file for the document that is initially opened. This also means that interactive documents which are only opened via other documents need not have their own information files.

SOME THINGS TO TRY

Before trying anything in this section you should have succeeded in getting SAMPLER.TBK to work by following the procedure described in README.TXT otherwise the procedures described in this section may not work.

Compiling a document

The document called YOURS.IDC is an exact copy of SAMPLER.IDC which you can edit and compile. It is suggested that before editing it you merely compile it. This will introduce you to some of the features of IDC. Follow these steps:

(1) Using a text editor (or a word processor that can save files as plain text) change the file RESOURCE.IDX to be compatible with your system. As supplied RESOURCE.IDX contains:

```
<!-- Resources for IDC Sampler 1.0 17th December 1993 -->

<!entity sampler "sampler:\">
<!entity images "c:\ken\idc\sampler\images\">
<!entity windows "windows:\">
<!entity toolbook "toolbook:\">
<!entity example "example:\">

<index>
<image name="Great Hall" file="&images;gh16.bmp">Great Hall</image>
<image name="C16" file="&images;gh16.bmp">Great Hall 4-bit (16 colours)</image>
<image name="C256" file="&images;gh256.bmp">Great Hall 8-bit (256 colours)</image>
<image name="Town Hall" file="&images;townhall.bmp">Leeds Town Hall</image>
<image name="Dales" file="&images;wharfdle.bmp">View of Wharfedale Beamsley
Beacon</image>
```

```
<image name=Country file="&images;ilkleymr.bmp">Walkers on Ilkley Moor</image>
<image name="Park" file="&images;canalgd2.bmp">Canal Gardens Roundhay</image>
<image name="Map" file="&images;gbsmall.bmp">Map of GB showing Leeds</image>
<image name=graph file="&images;veltime.bmp">A simple velocity-time graph</image>
<image name=amphetamine file="&images;amphet.bmp">Chemical structure of amphetamine</image>
<idc name="IDC Example" file="&example;example.tbk">A small example document</idc>
<app name=winhelp file="&windows;winhelp.hlp">Windows Help File</app>
<app name=notepad file=notepad.exe arg="&sampler;readme.txt">Notepad
Example</app>
<app name=Toolbook file="&toolbook;tbook.exe" arg="&toolbook;pages.tbk">Toolbook Example</app>
</index>
```

The only line which must be changed is the one which decares the entity **images**. This should be changed so that it's value is the path for the directory containing the *.BMP files supplied with the Sampler. Note that the path must include the final \.

(2) Open IDC (that is the compiler IDC_136.TBK; remember it must be opened with TOOLBOOK.EXE and SYS136.TBK must be in the same directory as IDC_136.TBK.)

(3) Wait for the Control Page to appear and then click Options.

(4) Load the options file SAMPLER.OPT. Follow these steps:

(4.1) Click the **Load Options from file** button.
(4.2) In the dialogue box type the full path and name of SAMPLER.OPT
(4.3) Click OK in the dialogue box

(5) You should see that some of the options have been set and boxes filled. It is necessary to change the path names for the Default File and for the Resources Index. Follow these steps:

(5.1) Click the Set Default File button.
(5.2) In the dialogue box type the full path and name of YOURS.IDC
(5.3) Click OK in the dialogue box
(5.4) Click the Set Resources Index button.
(5.5) In the dialogue box edit the full path and name of RESOURCE.IDX so that it is correct for your system.
(5.6) Click OK in the dialogue box

(6) Click OK on the Options page to return to the Control Page. Your options will have been saved by the compiler and will be loaded automatically next time it is opened.

(7) Click the Compile button. The file specification in the dialogue box should be the full path and name of YOURS.IDC. [If not repeat 5.1 - 5.3]

(8) Wait while YOURS.IDC is compiled. There should be no IDC errors at this stage.

(9) When the cursor has become an arrow again, click on the Detach button. The path and name in the dialogue box should be that of YOURS.IDC but with the IDC replaced by TBK. Click on the OK in the dialogue box. IDC will detach the new document and close.

(10) Testing out the new document.

(10.1) Check that a copy of COMM136.TBK is in the same directory as YOURS.TBK.
(10.2) Make a copy of the file SAMPLER.INF, call it YOURS.INF and place it in the same directory as YOURS.TBK.
(10.3) To see that your new document functions like its twin (SAMPLER.TBK) open the new document called YOURS.TBK. You may use any of the following methods:

(A) Create an icon in an appropriate program group and type in the following information:

Description: Yours
Command line: TBOOK *path*\YOURS.TBK

This method assumes that the directory containing TBOOK.EXE is mentioned in the PATH environment variable.

(B) If TBK is associated with TOOLBOOK.EXE or TBOOK.EXE then open the File Manager and double-click the file YOURS.TBK

(C) Open Toolbook and the open YOURS.IDC through Open on the File menu.

Note: If YOURS.TBK (or any Interactive Document) is open with the full ToolBook system then there is an extra option on the Actions menu. This option is Author and choosing moves into authoring mode. This will allow you (if you wish) to examine and even change the document. Changing it is a waste of time as re-compiling it will remove the changes. However, if you select the option Author from the Actions menu you will enter authoring mode and there is an additional menu to those normally available. This menu is IDC and only has one option which allows you to change the password which a user needs to know in order to abort the interactive document using control-X. Using control-X by-passes all the procedures such as saving bookmarks and it is therefore not desirable to use it in normal circumstances. However, during development or installation on a new system there may be times because of errors there is a need to use control-X. The password for intereactive documents supplied in the Sampler Kit is **please**. If you do change the password then you must save the interactive document. You are not allowed to save it using the current file name (unless you know the password which is *not* the same as the password associated with control-X); you must choose the Save As option and choose a new file name.

Changing the header

This requires you to edit the file YOURS.IDC using Notepad or another text editor. [You may use a word processor if you wish but remember to save it as plain text.]

- Change the text between <title> and </title>.
- Add your name to the credits by inserting it between <credits> and </credits>. If you want it to start on a new line the put <p> in front of it.
- Change the date to today's date.

When you have made all the changes you wish then re-compile the document by following steps (7) to (10) [except 10.2] of the section on Compiling a Document.

Adding a menu item and a new page

(1) Add a new page. This may be added at any point in the file YOURS.IDC immediately after a tag of the form </f?>. The obvious place to put it is just before </body>. Choose where you are going to type the new page. [Remember to use a text-editor or save the file as text.]

(2) At the chosen location type:

```
<f2 name=mine next=Menu>
<h1>My Page</h1>
This is a page that I have added to the IDC Sampler.
</f2>
```

(3) Now alter the menu. Find the part of the text which begins

```
<f1 name=Menu>
Illustrative Interactive Documents
<o go=MCQ>Multiple-choice Questions</o>
<o go=Test-Teach>Test-Teach Exercises</o>
<o go=Presentation>Presentation</o>
<o go=Hyperbook>Hyperbook</o>
<o go=Linking>Linking</o>
</f1>
```

and immediately before </f1> put in the line:

```
<o go=mine>This is my option</o>
```

save the edited file and then re-compile YOURS.IDC following the steps (7) to (10) [except 10.2] of the section on Compiling a Document.

If any errors occur during compiling, quit the compiler by clicking Quit and then check that your changes are exactly as prescribed.

Using a different format for the menu

The file BUTTON.TBK contains a format (called Button) which requires the same markup as the Menu format which is suppied as part of IDC. This format can be added to IDC and used as an alternative to the Menu format. To do this carry out the following steps:

(1) Add the Button format to IDC

(1.1) Open IDC
(1.2) Click the NAME button and in the dialogue box type:

Sampler 1.0 (Modified)

this will be the name of your new compiler.
(1.3) Click OK in the dialogue box.
(1.4) Click the ATTACH FORMATS button.
(1.5) Follow the instructions carefully. They are repeated here.
(1.5.1) Choose the Toolbook option in the Import dialogue box.
(1.5.2) Locate the file BUTTON.TBK in the usual way and then click OK. The format will be added to the compiler.
(1.5.3) Give a name for the compiler. [Creating a compiler with the same name as the current one is password-protected. Without the password you must choose a different name.]
(1.6) Create a program item for your new compiler.

(2) Re-assign formats and then compile YOURS.IDC with the new compiler.

(2.1) Open your new compiler.
(2.2) Click the Options button
(2.3) Click the Load Options file and load the options file SAMPLER.OPT - remember to give the full path and name.
(2.3) Click the name Buttons in the list of formats.
(2.4) Click button f1
Now f1 (which is the tag used with the markup for the menu) is now associated with the Buttons format.
(2.5) Click OK to return to the Control page
(2.6) Compile and test out YOURS.IDC as described in steps (7) to (10) [except 10.2] in the section on Compiling a Document.

Creating your own document

By modelling a document on SAMPLER.IDC or YOURS.IDC you can create your own interactive documents. It is not necessary to use all the formats, or to have a menu. If you wish use your own images (in bitmap format), link documents to other documents, or invoke other applications then you will either have to add the necessary declarations to the resource index or create a resource index of your own.

Showing an interest

IDC (as supplied) is fully functional, but without further information it would be very difficult to create new formats for IDC . To do this you need to be able to work at authoring level with the compiler (or a shell created from the compiler). Documentation for IDC is in preparation and copies will be made available in the near future. As yet no policy on the distribution of IDC has been determined, but it would be helpful if those who are interested in using it or wish to hear of further developments would send their names and addresses (at least) to:

K Tait
Computer Based Learning Unit
The University
Leeds LS2 9JT
Tel: +44 532 334628
Fax: +44 532 334635
Email: ken@uk.ac.leeds.cbl *or* ken@cbl.leeds.ac.uk