

Appendix 3 - Inter-Publish Documentation

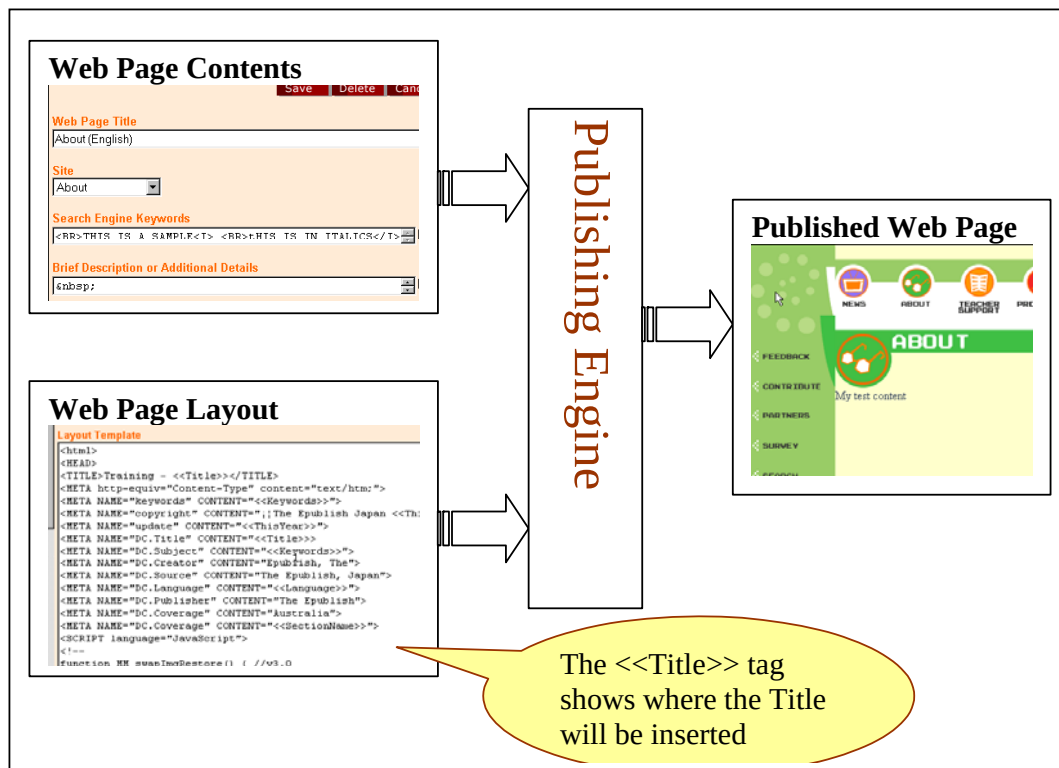
Supporting a widely dispersed end user population using the prototype software, extensive documentation was required. Two principle resources were provided to the end users: a training manual that provides simple authoring instructions and a designer's manual.

The designer's manual was developed to provide guidelines for those users who were working with the document ontology of the prototype and with the Inter-Publish Transformation Language. This appendix presents relevant extracts from the designer's manual relating to the Inter-Publish Markup Language and the document ontology editor for document types. The training manual, designer's manual and system administrators guide are available at http://www.interpublish.com.au/interpublish_documentation.html.

A3.1 Inter-publish Transformation Language (IPTL)

Inter-Publish allows the separation of content and design. The mark-up language in Inter-Publish provides the essential elements for content separation. The authors content is captured in forms whose fields the author can define the themselves: these are **Document Types**. When this content is published it is generated by merging the authors content fields into designated places in a **Layout**. A document may have many presentational layouts. The publishing process operates in a manner very similar to a mail merge, either on a single document or many, to bring many documents together in a report, index or mail out.

The mark-up language quite simply defines a set of generative rules for insertion of the authors **Content Fields** into specified points in the authors **Layouts**.



A simple web page layout might look something like:

```
<html>
<head>
<TITLE>Montage World<<Title>></TITLE>
<META NAME="keywords" CONTENT="Montage World, projects, schools, teachers, students, about">
<META NAME="copyright" CONTENT="&copy; The British Council of Australia <<ThisYear>>">
<META NAME="author" CONTENT="">
<META NAME="update" CONTENT="2000">
<META NAME="DC.Title" CONTENT="<<Title>>">
<META NAME="DC.Subject" CONTENT="<<Keywords>>">
<META NAME="DC.Creator" CONTENT="British Council, The British Council of Australia">
<META NAME="DC.Creator" CONTENT="">
<META NAME="DC.Source" CONTENT="The British Council of Australia">
<META NAME="DC.Language" CONTENT="English">
<META NAME="DC.Publisher" CONTENT="The British Council">
<META NAME="DC.Coverage" CONTENT="The British Council of Australia">
<META NAME="DC.Coverage" CONTENT="MontageWorldAbout">
</head>
<body background="images/bgtilsub.gif" bgcolor="#ffffff" text="#000000" link="#CC0000" vlink="#cc0000"
alink="#666666" marginheight="0" marginwidth="0" leftmargin="0" topmargin="0">
<table cellpadding="5" cellspacing="0" border="0"><tr><td>
<<Document>>
</td></tr></table>
</body>
</html>
```

A callout bubble points to the `<<Document>>` tag, stating: "The <<Document>> tag".

IPTL Mark-up Conventions

The syntactical conventions for transformation of text content units are very easy. Each field in an Inter-Publish form has a ***fieldname***. Substitution is generally in the format:

```
<<fieldname>>
```

For example, to substitute the **Title** field simply requires a tag:

```
<<Title>>
```

Similarly, the web page contents are generally inserted using:

```
<<Document>>.
```

Additional tag formats are:

Tag	Function
<<fieldname>>	show the contents of the field
<<Upper_elementname>>	show field contents in upper case
<<Lower_elementname>>	show the field data in lower case
<<Strip_elementname>>	show data stripped of special characters and with spaces removed
<<Split_elementname>>	show the data with all special characters removed and spaces replaced with " _ "
<<HTMLEncode_elementname>>	apply html encoding to the contents of the field
<<Mailto_elementname>>	insert the text mailto: in front of an email field if none is already present
<<URL_elementname>>	insert the text http:// in front of a URL field if none is already present
<<Flash_elementname>>	strips html encoding that is not compatible with flash-format html content
<<StripBR_elementname>>	Removes tags and line feeds to make a single line of text
<<XMLEncode_elementname>>	Show the data XML encoded
<<Count_elementname>>	A tabular total of survey occurrences for a particular document
<<Image>>	Location to insert the binary contents of the Image data

These tags define the point within static content that database or other information is to be inserted. There are some tags designed for specific use in indexes and in document publishing. Others insert static information such as current date or time.

Document Layout Tags

All document field elements have an associated tag. These can be reviewed from the Types Menu in System Administration. Each Element is referenced simply by its name. For instance, a standard document might comprise the following elements:

Title

Description
 Document
 ImageReference (shows associated **images** attached to this entry)
 LinkReference (shows associated **link** documents attached to this entry)
 PublishArray
 SubjectName (a repeating field)

The following table lists the common field names which can be used when designing a Document Layout.

Information from the Document or Document Collection will automatically be inserted at each occurrence of these tags in the layout. Commonly used Tags are listed below. You should review the fields available within the specific document type being published for a full list of tags available to a given document type. The author can create the authors own document types and fields.

Tag	Function
<<DocumentID>>	Document identifier
<<DepartmentName>>	Name of the Department to which the document belongs (and its heading in the Table of Contents)
<<SectionName>>	Name of the Section to which the document belongs
<<Title>>	Title of the document
<<Author>>	Last Author of the document
<<TimeCreated>>	Date/Time the document was created
<<DocumentStatus>>	Published statusPublished, Draft, Archived, etc
<<PublishURL>>	Published URL for the document
<<PublishName>>	Show only the name portion of the URL
<<PublishingDestination>>	Insert the URL path of the page
<<Description>>	Document abstract/summary
<<Document>>	Full document or web page details
<<DocumentTable>>	Format the document automatically as a table, replacing TAB characters with column tags.
<<Keywords>>	Free-form keywords
<<Today>>	Day of the week
<<ThisYear>>	The current year in yyyy format (for copyright and other uses)
<<ThisMonth>>	The current month number
<<ThisDay>>	The current day number
<<ThisMonthName>>	The current month name
<<ThisHour>>	The current hour
<<ThisMinute>>	The current minute
<<MetaDescription>>	The contents of the Description element with html tags removed and truncated to 200 characters
<<tab>>	Inert a "tab" chr(9) character
<<DcumentTable>>	Format the document element converting tab rows to a table
<<xmlnamevalue>>	Insert a XML name-value pair set of all elements in the document
<<autoform>>	Create an HTML maintenance form for the current document

The title would be referenced in a layout as <<Title>>. The subject name, being a repeating field in Montage, would be referenced as follows:

<<Repeat_SubjectName>> <<SubjectName>> <</Repeat>>.

In addition to the normal document fields, the following have specific functions:

<<NextTitle>> Substitutes with the NEXT **published** document id in the document id sequence (for the Program to which the document belongs) from the currently displayed document.

<<PreviousTitle>> Substitutes with the PREVIOUS **published** document id (for the Program to which the document belongs) in the document id sequence from the currently displayed document.

<<AuthorName>> Insert the name of the last content editor.

Date Tags

Where you have date input fields there are several date presentation tags which you may use to format this date information in a specific way. *Note:* these are at present limited to English-language presentation of dates:

Tag	Description
<<DayNumber_elementname>>	Show only the day of month number from the nominated date element. Leading zeroes are suppressed.
<<DayNumber2_elementname>>	Show only the day of month number from the nominated date element. The format of display is “##” – eg 02. Leading zeroes are suppressed.
<<MonthNumber_elementname>>	Show only the day of month number from the nominated date element
<<MonthNumber2_elementname>>	Show only the day of month number from the nominated date element. The format of display is “##” – eg 02.
<<YearNumber_elementname>>	Show only the year number from the nominated date element
<<MonthName_elementname>>	Show the English equivalent of the month name (in full) – eg “July”.
<<DayName_elementname>>	Show the English equivalent of the day of week name (in full) – eg “Monday”.
<<Today>>	Day of the week today.
<<ThisYear>>	The current year in yyyy format (for copyright and other uses)

Conditional Tags

Conditional tags control the display of blocks of static content and replacement tags based on the presence or absence of information within Inter-Publish data fields. The syntax is:

```
<<HasValue_Field>> {contents to be shown} <</ HasValue_Field>>
```

```
eg <<HasValue_Description>> <b>Brief summary<br></b> <<Description>>
<</HasValue_Description>>
```

The above example would display a heading of "**Brief Description**" followed by the contents of the "Description" field ONLY if the Description Field contains some text.

To show content only where a value is *absent*, The syntax is:

```
eg <<HasNoValue_Description>> No Information Supplied <</HasNoValue_Description>>
```

The above example would display the message "No Information Supplied" if the content of Description has no value. To test for a *particular* value the following syntax is available:

```
<<contains="valuetotest" field>> (contents to be shown) <</contains="valuetotest" field>>
```

For example:

```
<select name=fruit>
<option <<contains="banana" field>> selected <</contains=" banana" field>> value="banana">banana
<option <<contains="apple" field>> selected <</contains=" apple" field>> value=" apple"> apple
</select>
```

All the standard Inter-Publish tags can be qualified by conditional tags.

Including other Documents and Indexes

The following tag will include and process the contents of any other document:

```
<<Include "SectionName!!Title of the document">>
```

eg

```
<<Include "Mysite – Includes!!Feature Feedback Item">> or just
<<Include "Feature Feedback Item">> or just
```

Emailing Documents

Documents may be emailed. To send a document via email you must configure the document with the following:

- a “publishing destination” of `mailto:servername` where `servername` is the name of the mail server which will accept the authors email
- a “publishing url name” which specifies the ELEMENT to be used as the email address
- a layout formatted as an email, eg:

```
Dear <<ContactName>>
The authors message
Regards
<<AuthorName>>
```

Indexing and Transforming Collections

In some cases you may wish to transform a collection of document or content units. Inter-Publish has two features which support this process: an indexing engine which allows you to specify complex persistent searches based on complex rules; and the Transformation language, which allows the results of a selection process to be presented in a new style or form, with potentially multiple layers of presentation of the results of an index.

Many *field elements* in a document can have repeating contents. For example, a document might have many subjects. In a Layout, the tag to show all subjects for a document could appear as:

```
<<Repeat_SubjectName>> <<SubjectName>> <</Repeat>>
```

Documents can link to other content items (including Documents) creating a hierarchical structure. The implicit hierarchy for publishing Documents in Inter-Publish is:

Collections *which have*

Documents *which have*

Elements

Both document publishing and indexing allow repeating content to be published in a structured manner. Similarly, the Inter-Publish Indexing function allows dissecting of information within Inter-Publish according to user-defined hierarchies of content, for example:

"Create a subject index for all documents in the Section called *History of Multimedia*.

The general syntax for repeating information is:

```
<<Repeat_itemtorepeat>> HTML content to be created <</Repeat>>
```

The following repeat structures for indexing describe a non-nested repeat loop:

```
<<Repeat_Category>>
<<Item>>
The HTML to be repeated for each document item within the category
<</Item>>
<</Repeat>>
```

A special form of Indexing repeat structure allows the construction of a list of repeating items in tabular form (for example a table of survey responses).

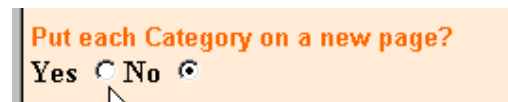
```
<<Repeat_Array>>
  This HTML will repeat for each DOCUMENT within the collection
  <<Item>>
    Each primary sort CATEGORY will be repeated for EACH document
    <</Item>>
  <</Repeat>>
```

Indexing Features

Indexing can generate web pages or emails across part or all of the authors content. The Primary Sort Category serves several purposes in indexing:

- it governs the order in which results will be displayed
- it governs when sub-pages will be created if you want each category on a separate page.

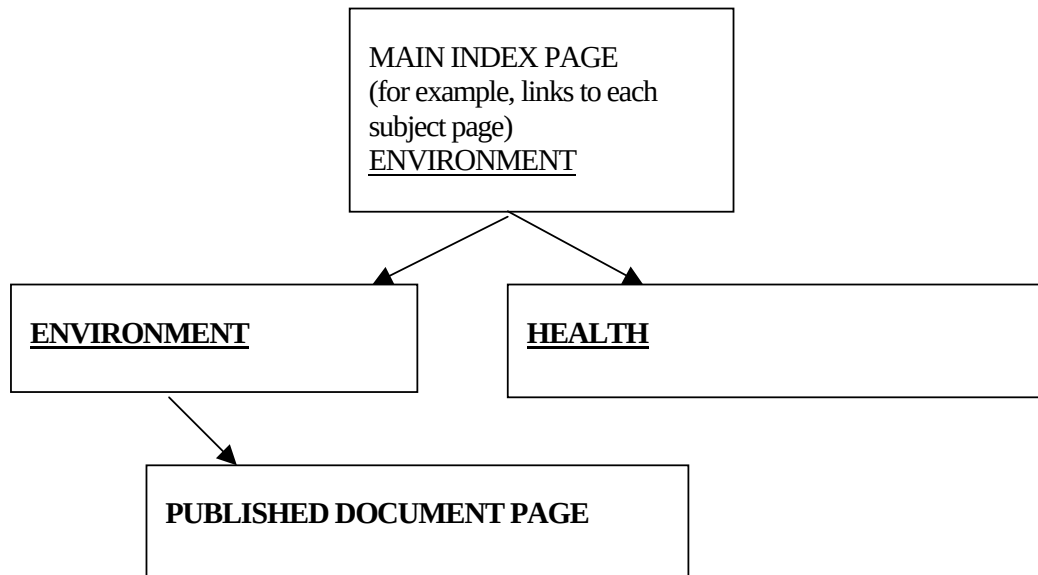
The use of sub-page indexing allows you to generate a series of pages automatically for example a sub-page for each "subject". To use this function just click **Yes** on the following radio button:



Put each Category on a new page?

Yes ☒ No ☐

All web page contents within the <<Repeat_Category>> statement will be created on a separate page named as follows:



The author can over-ride the default sub-page naming by supplying a sub-page prefix:

Index Sub-Page Prefix (use in name ## for item sequence rather than category in suffix)

For instance, in a subject index with a sub-page prefix of "subject_" the web pages may be created as:

subject_environment.htm

subject_science.htm

If the author also specify ## in the prefix name, a sequential ordering of pages will be created with no suffix, for example: event##.txt will create:

event1.txt

event2.txt

...and so on

Remember the author can use the <<include>> tag to include indexes in other documents, or the contents of documents in other indexes.

Indexing Tags

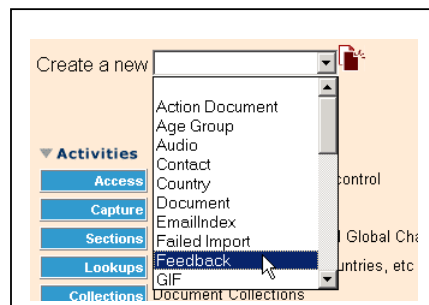
The following tags are available only within the <<Repeat_Category>> structure.

Tag	Description
<<category>>	Show the current sort category value
<<ItemCount>>	Show the total number of items within a given sort category
<<indextitle>>	Show the title of the index document
<<Indexshortdescription>>	Insert the contents of the index “short description” element
<< Indexdocument>>	Insert the contents of the index “short document” element
<< indexurl>>	Show the URL of the index (as distinct from the particular document beingindexed)
<<IndexPublishName>>	Show the url name only of the index
<<IndexPublishDirectory>>	Show the URL path of the index
<<ItemNo>>	Insert an incrementing number within category as a count of documents starting at 1
<<Alternate1>>value1<</Alternate1>> <<Alternate2>>value2<</Alternate2>>	Value 1 and value2 will appear in odd/even alternates (used to build a two column table)

Proforma Documents

In every section, and for every Document Type the author can define “Proforma” documents. These “Proforma” documents cannot be published. Using them you may define default values on each and every **element** of a Document Type, including publishing destinations. This is especially valuable if a given **document type** should publish to multiple destinations.

Creating a Proforma document is easy. Just create a new document of the **document type** for which you want defaults:



For example, to create a “feedback” proforma document, go to the Inter-Publish Home Page and select “Feedback” and click the CREATE button.

The title of the document **must** contain the term “Proforma”, eg:

Please supply the details below:

Feedback Title
Proforma Feedback

Language
English

Click on the save button. In the resulting form the author can enter defaults in any of the form values, including the publishing destination. The “Publishing Destination” may be configured to have a prefix or suffix of a particular sort (for example .html rather than the default of .htm):

Published Name

FB_ <<Title>>.html

La A

Once you save the proforma, it will become the default for all **documents** of the **document type** of the proforma.

Inter-Publish content headers

The Inter-Publish user interface can be tailored to handle language headers for multiple languages, with a language prompt at sign-on. If a particular HTML language encoding method is not yet registered in Inter-Publish, the author can define the relevant content headers using the following simple procedure. Select the “site setup” option from the Inter-Publish Home Page.



Then select the “Languages” option under “Site Structure”.

Site Structure	
Server Defaults	Client System Defaults
Languages	Languages

The author will see a list of existing registered languages. Only those with a defined HTML “content header” will be displayed at logon time. The “content header” defines the HTML encoding methodology for the language. If the author needs to define a new language entry, the authors system administrator should edit or add the relevant language code to add the content headers for that language.

#	ID	Full Name	Content Headers	Acti
1	en	English...	text/html;CHARSET=iso-8859-1	Edit
2	ja-euc	Japanese (Shift-EUC)...	text/html;CHARSET=x-euc-jp	Edit
3	ja-jis	Japanese (Shift-JIS)...	text/html;CHARSET=shift-jis	Edit

HTTP headers can be any standard HTTP header information. HTTP headers allow the program [and the web server] to send control and other information to the web browser which isn't displayed, but used to provide meta-information about the content which is about to be sent. The "Content-Type" header denotes what type of data is contained in the content as a MIME type. The CHARSET option defines the language encoding methodology. Note that some languages such as Japanese have multiple different encoding methodologies. You should select a particular encoding approach for a given site. Inter-Publish can, however, manage different encoding approaches on a document-by-document basis as long as the language encoding approach is defined for the language.

Other common content types are text/plain (for content returned as text instead of interpreted HTML statements), image/gif (for GIF images), image/jpeg (for JPEG images), video/quicktime (for movies in the Apple QuickTime® format), and text/xml (for XML documents). In addition, a Web server or Web browser may support custom MIME types. For instance, to see the content types already defined by a Microsoft Web server, use the Internet Information Services snap-in, to open the property sheets for the authors Web site, click the HTTP Headers tab, and then click the File Types tab. For more reading on MIME headers standards see the W3C http://www.w3.org/Protocols/rfc1341/4_Content-Type.html.

Content Interchange and XML

In addition to XML syndication and web services content interchange, Inter-Publish has a simplified XML content interface intended as a light-weight means of information interchange between the authors website and the content management system. Delivery of the content by email means that the content management system can reside on an internal intranet with no external access if security restrictions so require, while information content can be exchanged with the authors external web server using page publishing and email pickup of user content submissions.

The publishing software will accept content entries through email delivery from a standard POP account. The POP account to be used is specified in the Section set-up, on a section-by-section basis.

The publishing system is data driven based on the data fields passed to in a defined XML format.

Simple XML content interchange

A minimum set of elements needed to create a document in the publishing system comprises: *ClientID*, *Title*, *ContentType*, *DocumentType*, *SectionName* and *Language*. These minimally define the type of the document it is, the client destination for the document and the title and language of the document. A simple field (no repeating values) has the following format:

```
<ElementValue>
<Element>Name of the element</Element>
<Sequence></Sequence>
<Value>value of the entry</Value>
</ElementValue>
```

A field with many repeating values (for example Subject) will also have a sequence number:

```
<ElementValue>
<Element>Name of the element</Element>
<Sequence>00001</Sequence>
<Value>value of the entry</Value>
</ElementValue>
```

Where possible, the script to send the data should be generalized with all data elements passed from the HTML form as visible or hidden elements, excluding publishing destination, language, *ContentType*, *documenttype*, *clientid* and *title*, which are mandatory. Inter-Publish will create Document Types and Field Elements which do not already exist. This means that the author can create "document types" easily just by sending an XML content entry for moderation.

Moderation

When a content item is received from the POP account it is automatically parsed by Inter-Publish and a document of the relevant type is created. If the document type does not exist it will be established, and fields will also automatically be created. The new document will appear on the Inter-Publish main menu visible to users with access to the relevant section, and will be shown as "awaiting moderation."

Bypassing Moderation

If you wish an item to appear directly in the Inter-Publish without appearing in the moderation screen, pass a documentstatus of either Draft or Published:

eg SetXML ("DocumentStatus","Published","")

Autopublishing & XML updates

If you send an XML entry with an Element of "ExternalID" and a Title of "XMLUPDATE", the fields passed will update the contents of the entry described by ExternalID and auto publish this entry as long as a database match is found on the supplied ExternalID.

Document Types

Document Types and Elements define the basic units of information stored in Inter-Publish. It is important to understand this if the author is creating a new site. They are a means of simplifying and customising the user interface for content capture, and of simplifying and extending the layout design when it comes to publishing. They are fundamental to effective content reuse.

This Section will introduce methods for tailoring the content capture forms to meet the specific information needs of different users.

Document Types define the fields that are shown to a content provider who is editing content on the publishing system. There is no limit to the number of document types that can be defined. Examples of **Document Types** could include:

- Image
- Feedback
- School
- Registration
- Book

Each document type might be directed toward a particular type of information. A document type of *feedback* for instance might show the following items:

- Feedback Title
- Contact Name
- Contact Email
- Comments
- OK To Publish

Document Types are themselves categorised into Content Types, which bring together a range of similar content items. For instance, a single **Content Type** of "Web Pages" might contain **Document Types** of "Web Page", "Survey", and "Script" while a **Content Type** of "Image" might contain "JPG", "GIF", "Image", etc.

Document Types contain **Elements** (like Fields in a database or Columns in a Table). This section will show how to add and configure Document Types and Elements through the Inter-Publish user interface.

Creating Document Types

From the Inter-Publish **Home Page** click on **Site Setup** under the **Activities** heading. This will take you to the System Administration menu. The author will see a link **Types**. Click on this link and the author will see the Document Type management screen. Inter-Publish comes with a predefined set of Document Types and Field Elements. In many cases the author can simply use or copy existing document types and modify them to the authors needs.

#	Document Type	Language	Activities
1	Action Document	en	Copy Edit
2	Age Group	en	Copy Edit

To create a new Document Type simply click on the NEW button and:

1. Give the Document Type a name
2. Select a Content Type to which this document will belong. Where possible use an existing Content Type that is appropriate for the type of contents the author is dealing with.

Content Type conventions in Montage are:

- Document – for web pages
- Image – for all images and animations
- Audio – for audio content
- Video – for video content
- Feedback – for website feedback items

- Project – for website projects
 - Survey – for survey questions and responses
3. Select the Element Names you want to use. The **Prefix** and **Suffix** columns define the heading text and additional information the author can include on the Inter-Publish form for this document type. The Elements define the input format (a single line text box, a full text area, a yes/no radio button, etc), as well as possible lookup values.

Name	Prefix	Suffix
Title	 Title 	

4. Click on the SAVE button.

The author will find that Inter-Publish comes pre-loaded with a wide range of **Elements** to pick from. If you need to create new Field elements for use within a document type, this is done using the **Elements** link on the Setup menu. The components of a field element are:

Attribute Name:
Short Description:
Field Format:
Field Language (if required):
Field Repeats (1=once, -n=always show n new inputs, +n=show exactly n inputs):
Field Size (display size):
Field Size (maximum input):
Document Contents Field:
Field Values:
Attribute Of:

Attribute of the element	Attribute role
Short Description	A brief textual description of the element which can appear in help popups.
Field Format	How the element will be shown to the user. (see below)
Language	If this field is to use a specific language
Field Repeats	If this field is to allow multiple selections
Field Size	The maximum input textbox size
Field Maximum	The absolute maximum characters allowed
Document Contents Field	DocumentText – for free-form content MetaText1 – for content items < 240 characters in length The qualifier mandatory can be used to indicate that this element is required for publishing.
Field Values –	A comma delimited list of valid values where the Field Format has been selected as standard HTML. Select list used only for complex Element structures. Sub elements of an XMLArray should all point to their parent element.
Element Of	

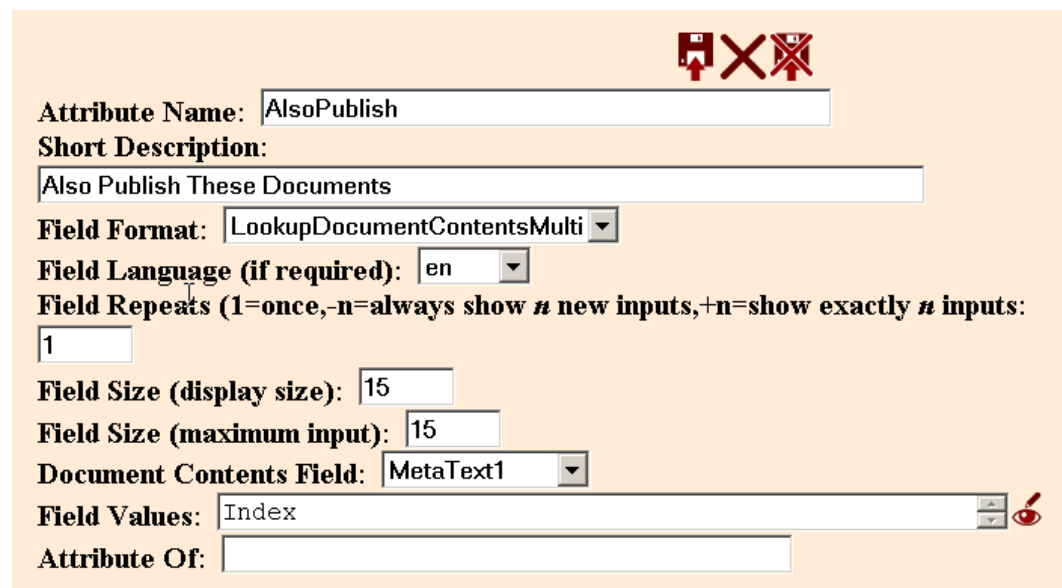
Content authoring formats are:

Format	Function
Date	Similar to Input, but with date validation of the contents.
FileUpload	Used to display a file upload option on the form
Hidden	The field will not be shown, but the information will be stored.
HTML	Displays an HTML “textarea” input box with auto-formatting of the html content (this is the format generally selected for HTML text input)
Input	Displays as a normal HTML “text” input box
InputNoPublish	Displays as a normal HTML “text” input box, but this item will not be included in some publishing functions
Logical	Displays as a Yes/No Radio button
LookupCollection	Displays a list of collections
LookupDivision	Displays a list of configured Divisions
LookupDocumentContents	Used to show a list of existing content items based on a selection rule supplied in the Field Values field. The format of the Field Values entry will be either: <i>DocumentType</i> which shows all entries of a particular document type or <i>SectionLimit DocumentType</i> – shows all entries of a particular document type limited to the section to which the document belongs.
LookupDocumentContentsMulti	Similar to Lookupdocumentcontents, but displays as a multi-select HTML select list
LookupElement	Used to show a list of Inter-publish Elements
LookupLanguage	Displays a list of registered languages
Numeric	Numeric input with numeric validation
Password	Displays the content typed in HTML “password” format (* shows for characters typed)
Reference	Displays as a content linkage item (for example Document linking to Images). This is the means by which relationships between documents can be defined (for example Schools might be linked to Teachers, Web Pages to Images). The format for the reference name is: <i>ContentTypeReference</i> .
Select	Displays a standard HTML select list based on the comma-delimited list defined in Field Values
SelectNoPublish	Similar to select, but some publishing functions will ignore this element
SysAdminInput	Display this field only for system administrators
TextArea	Similar to HTML, but without some of the XHTML formatting functions
TextareaNoPublish	Similar to TextArea, but this item will not be included in some publishing functions
TextAreaNoVisualEdit	Similar to TextArea, but with no VisualEdit button (used for scripts)
Thesaurus	Display a list of controlled vocabulary entries based on the controlled vocabulary defined in Field Values
ThesaurusMulti	As with the Thesaurus element, but allow multi-selection of a single select list
XMLArray	This is a “parent” element defining sub-elements that will appear as a repeating array of elements (such as a publishing

	destination). The Field Values must contain the XML definition of the elements to be included, in the format: <pre><ElementList> <ElementValue><Name>elementname</Name></ElementValue> <ElementValue><Name>elementname</Name></ElementValue> <ElementValue><Name>elementname</Name></ElementValue> </ElementList></pre> For example, to show “EventLocation”, “EventStartDate” and “EventFinishDate” as a repeating array of elements, use the following definition: <pre><ElementList> <ElementValue><Name>EventLocation</Name></ElementValue> <ElementValue><Name>EventStartDate</Name></ElementValue> <ElementValue><Name>EventFinishDate</Name></ElementValue> </ElementList></pre>
--	---

AlsoPublish

The author can add a special element **AlsoPublish** to any document definition.



Attribute Name:

Short Description:

Field Format:

Field Language (if required):

Field Repeats (1=once, -n=always show *n* new inputs, +n=show exactly *n* inputs:

Field Size (display size):

Field Size (maximum input):

Document Contents Field:

Field Values:

Attribute Of:

The AlsoPublish element can be used to allow the selection of an Index or associated document that should *always* publish when the currently selected document is published. This can be used to enforce synchronisation of documents & particular indexes.

Standard Document Types

Those who are supplying content in Inter-Publish work with content capture screens based on Document Types which define the structure of information captured in Inter-Publish. These

document structures assist later re-use of this information. The Document Type defines the fields appearing on these screens. The content editor can select (and change) which document type. Document Types can also be used to categorise and organise the website content. A useful convention for creation of web pages illustrated in the Montage site is:

Document or Web Page: a Document Type representing a normal web page that will appear in the sitemap

Action Document – representing a web page not appearing in the site map (for example feedback thanks). Some content pages are useful subsidiary pages that may also be irrelevant for the site map.

Script – a script or page that does not allow use of the visual editor (for example text input into a flash program). Inter-Publish comes with a wide range of established Document Types, which are suitable for most web publishing activities. The next section describes in detail how to establish the authors own document types.

Following such a convention allows the simple generation and update of a site map for the authors site. The section on Customisation of Document Types below for more information regarding the creation of the authors own **Document Types**.

The Montage "White" Site

The Montage "White Site" provides a basic set of website features to accelerate the implementation of a public web site. Site replication features allow you to take design the structure of the authors website and map features of the Prototype site to this structure.

The prototype site contains the following website sections:

Home – a navigational home page – the entry point for access to the key features of the site. This model home page provides a launch point to each of the following features.

About – the "introduction" section – will generally have organisational details and the site map.

Calendar of Events – a facility for the publication of forthcoming events and an event notice subscription facility.

Feedback – for the management of an online customer feedback section of the website. Facilities include the ability to highlight "feature" feedback items.

Linkup – for a public, moderated, linkup system. Clients can register online. Scripts are provided for automatic registration on Webboard.

Links – an online link recommendation system for the construction of moderated portal sites.

Projects – a system for the publishing online curriculum projects or to present searchable information on organisational projects or products.

Register – a system for moderated customer registration. Can be used as a portal to content for registered access only.

Search – a model search engine configuration using Microsoft Indexing Server for the authors site.

Survey – a simple survey system. Online moderated surveys.

Of course, the first stage in website design is a thorough review of requirements. Storyboarding and mapping the site design, and even some prototyping using a standard web authoring tool can be advantageous to assist development of an effective site structure. It may help the design evaluation to visit sites that illustrate features that may be included. The following are some sites worth visiting which have examples of different website features:

Home page site structure: <http://www.e-linkschools.org/>

Feedback – <http://www.montageplus.co.uk/feedback.htm>

About – <http://www.montageplus.co.uk/about.htm>

Links – <http://www.montageplus.co.uk/contribute.htm>

Projects: <http://www.montageplus.co.uk/projects>

Linkup & registration: <http://www.montageplus.co.uk/commonlink/register.htm>

Search: <http://www.e-linkschools.org/english/search/search.htm>

Survey: <http://www.e-linkschools.org/english/survey/survey.htm>

Once a site structure is conceived, it can be implemented very simply in Inter-Publish. The site structure in Inter-Publish should mirror that of the web-site.

Checklist for building a site from the Prototype

The following checklist may be useful in the establishment of designs for a new site. Designers unfamiliar with Inter-Publish may wish to review the Detailed Example which appears after the checklist, and should also review the Inter-Publish training manual for more information on Editing, Publishing and the use of the integrated Visual Editor.

	ACTION	
1	Before you proceed with Inter-Publish Section Set-up the author will need: • Be familiar with navigating and using Inter-Publish features • To have created the relevant directory on the authors external web server • To have the ftp access details for that directory • Have a System Administrator account on Inter-Publish	☐
2	If this is a new website, the author will need to create a new Web Site definition. This appears as the main heading for the site on the left hand navigation bar. From the Main Menu click on <u>Site Set-up</u> . Click on the <u>Web Sites</u> link. Click on the NEW button to create a new website Change the DivisionID and Division Name to a descriptive name for the Webiste. Click on the SAVE button.	☐

2	Create a Section entry for each sub-site. From the Main Menu select Section from the drop-down list and click on the CREATE button. The Section Name should generally be descriptive of both the Web Site and Sub-site it belongs to: for example if my web site is called "The Great Dividing Range" then my Section might be called "The Great Dividing Rangeabout". Supply here the base URL for the Site or Sub-Site Specify the physical location of the authors staging_directory. The author will see the Section Set-up page.	☐
3	Click on the browser REFRESH button. You should see the new website and sections appear on the left hand navigation panel.	☐
4	Against each new sub-site you should now see a Pub/Dest. The Publishing destination specifies the specific security configuration (usually ftp) for a particular subsite. Click on the <u>Pub/Dest</u> link and click <u>Edit</u> to enter the required publishing details. Supply the Server Name, UserID and Password for FTP access to each site in the Publishing Destination entry.	☐
5	Well done! The authors site is ready for publishing. The remaining stages involve the definition of the authors graphical design templates. The system has a prototype site which the author can use to "inherit" a set of useful site features. At this stage the author can use the site replication features to copy across these features to the authors new site. To use the site replication features, click on <u>Site Set-up</u> link from the Main Menu. Click on the <u>Replicate</u> link. The author will be asked to map the source and destination Clients, Web sites and sections. All contents will be transferred from the source to target sections, with URL's and other text changes applied.	☐
6	The authors graphical designers should now establish default design templates for each section. The layouts for each section can easily be reached simply through opening the Site List. If the author has replicated a prototype section the author will already see Layouts for each section. Click on the Layout link and on the right-hand frame the author will see a list of Layouts available for the Section. Edit each layout and apply the authors HTML design to the basic layout design presented. Be careful to preserve scripts and Inter-Publish markup tags. More details on the markup language is provided in Section 3 of this manual. More information on using the visual editor and publishing features can be found in the Inter-Publish training manual.	☐
7	Once the graphical designers have created the default templates, you should register these defaults in the Section configuration for each sub-site.	☐

Metatag Standards

Template inheritance allows inclusion of standard tags within every page. Every published web page that will appear in the Site Map or needs to be reachable by a search engine should contain the following Metatag entries (Dublin core DCMI metatags). These can be copied from the Prototype site documents or inherited in the template.

```
<TITLE><<Title>></Title>
<META NAME="keywords" CONTENT="<<Keywords>>">
<META NAME="copyright" CONTENT="&copy; The authors Organisation <<ThisYear>>">
<META NAME="update" CONTENT="<<Today>>">
<META NAME="DC.Title" CONTENT="<<Title>>">
<META NAME="DC.Subject" CONTENT="<<Keywords>>">
<META NAME="DC.Creator" CONTENT="<<ClientID>>">
<META NAME="DC.Creator" CONTENT="<<AuthorName>>">
<META NAME="author" CONTENT="<<AuthorName>>">
```

```
<META NAME="DC.Source" CONTENT="<<ClientID>>">
<META NAME="DC.Language" CONTENT="<<Language>>">
<META NAME="DC.Publisher" CONTENT="<<ClientID>>">
<META NAME="DC.Coverage" CONTENT="Australia">
```

Content Capture with the FTP Interface

Limitations

- To use this utility you must install the Inter-Publish FTP Daemon during installation (refer to the Inter-Publish System Administration Guide).
- This utility is targeted for content upload and maintenance, but not for site creation. The author cannot create directories and subdirectories, sections and subsections using the FTP interface. That must be done online using the Inter-Publish User Interface.

The authors system administrator will advise you of the FTP Server address for the authors site.

The author will need to be a registered Inter-Publish user with relevant access permissions. The author will only see the sections to which you have access.

Logging On

The format for the user id for Inter-Publish access is:

Client Name|Inter-Publish UserID

For example “WorldFirstWeb|ebalnaves” would be used to connect to the client configuration of “WorldFirstWeb” with user id “ebalnaves”.

You should enter the authors normal Inter-Publish password.

Multiple images can be selected for upload. They must still be “published” through Inter-Publish to appear on the target website. This is a very effective way of building an image library with minimal effort.

New HTML content, which is uploaded, will automatically:

- Create a content entry
- Put the HTML in a Layout associated with the content entry
- Scan the HTML content for title, keywords and Description metadata
- Javascript, ASP script and CSS entries will be uploaded as a document type of **script**.

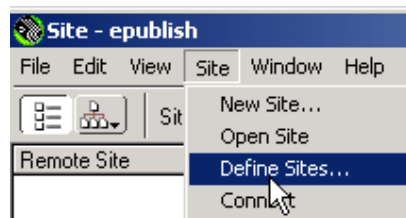
Microsoft FTP Client

The command-line client of Microsoft FTP. The authors system administrator will advise you of the FTP Server address for the authors site.

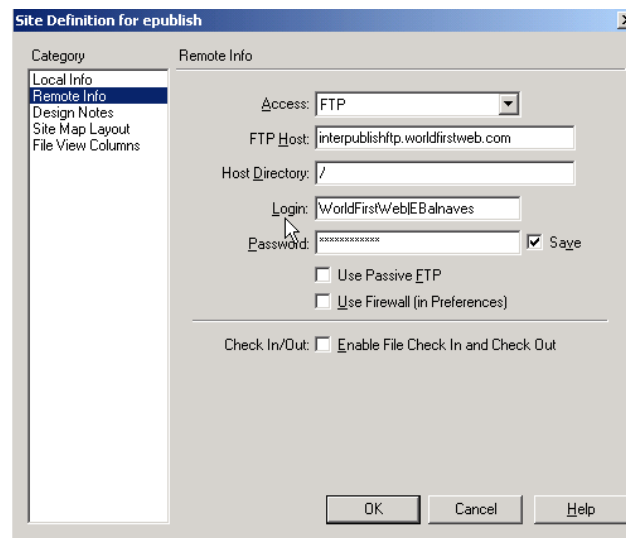
Dreamweaver Configuration

Start Dreamweaver.

Select **Site** then **Define Sites** from the Toolbar:



Enter the authors FTP access details



Use Site Connect to connect to the server. The author will then see a list of URL destinations to which you have publishing access.

The Database Import Utility

Limitations

- If the author has an access "Linked" database, you should load the "linked" table database (containing the actual data) not the controlling database.
- The author can only load one table from a database at a time
- Table relationships are not automatically created at this time

Features

- Automatic creation of new Document Types and Elements
- Selection of closest Element matches where possible.

The database import utility allows you to load a Microsoft Access table, mapping it to an existing or new Document Type in Inter-Publish. The facility is available under site set-up:



Then select the option “Import” under “Document Management”

Document Management	
Remove Span tags	Remove span tags from web page
Recover	Recover a deleted document by URL
Delete by URL	Delete E-publish entries by URL
Failed Imports	Review failed moderation items
Capture	Download images/layouts from a :
Ftp Queue	Failed Ftp Queue Management
Import	Import from an ODBC source
Generate a publishing	Allows the creation of a standard

The document import definition page will appear:

Access & Dbase files must be uploaded in order to be processed. If you have not already uploaded your file, browse the file now and select **UPLOAD**

Process the following file

Name of the table to import

Document Type
- the specific content Form to be used.

Content Type
- the generic content category
eg Document, Image, Contact.

Create in Section:

Step 1: The first step to import a database is to upload the database to the content server using the **BROWSE** and **UPLOAD** buttons.

Step 2: The second step is to define the name of the table to import and how this table will be mapped to Content Types & Document Types. The author can nominate existing or entirely new document types. If you create a new Document Type, the Document Type definition will be created

for you automatically during the upload. You must also nominate the Section into which the content will be loaded.

Step 3: The author will be asked to *map* the fields/columns in the source table to the **elements** to be used in Inter-Publish. Again, the author can ask Inter-Publish to create the new elements based on the table column names if you wish.

Persistent Search Collections

Once a site achieves a large number of web pages, images and other content to manage, it is useful to access, organise and publishing this valuable content in a number of different ways.

Collections and **Indexes** provide a means of easily bringing together content. A Collection is simply a set of documents that are selected by a flexible set of rules. Collections can be used to:

- Search for and edit documents that the author frequently needs that are across several sections. For example – show all documents of a particular document time (for example customer registrations across all sites); show all documents matching with or without a particular element value (for example, show all **customer** documents belonging to a particular state or country).
- Collections can be used to build indexes of the authors website content in mail-merge-style web or email publishing – for example, emailing all users who have provided feedback.
- Global Change operations to website content (for example, changing all references from one URL to another) can be limited to just those documents referenced by a collection.

Collections are easily defined, and operate on a simple set of criteria, which you specify and are persistent (they can be reused). Indexes are very powerful, but more complex to specify. The Indexing system is like a report writer or mail merge system. It can use Collections to select from across the breadth of the authors content, and then publish web pages (for example Subject Indexes, Site maps) or Email mail-outs based on the authors requirements. Several sample indexes are

supplied. Indexes have a meta-language for advanced selection and design that are described in more detail in the E-publish Designers manual.

1. From the **Home Page** select the **Collections** link under the **Activities** menu.



3. Click on the **NEW** button to create a new collection. Give the collection a descriptive name, for example: **Feedback items without a country**. Select a section to which the collection will belong.

 A screenshot of a form for creating a new collection. It has two main fields:

- Collection Name:** A text input field containing the text 'Feedback Items without a Country'.
- Section:** A dropdown menu with 'Feedback' selected.

4. The author is now ready to select the documents that should appear in you collection. The drop-down list under **Select Sections** and **Select Document Types** allow you to specify one or more Sections and Document Types to limit the authors collection. In this case, just select **Feedback** under **Document Types**. This will have the effect of selecting out only Feedback items.

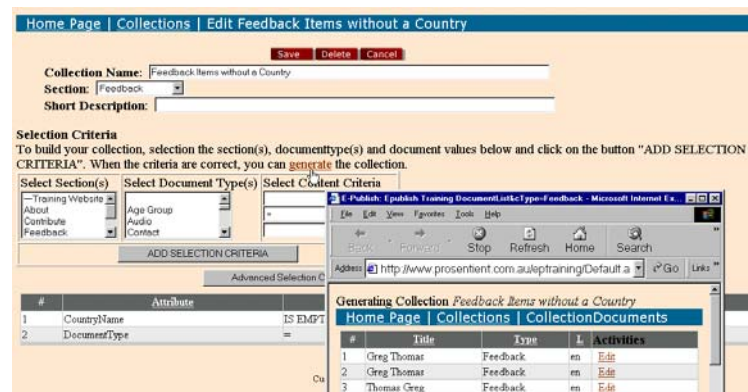
 A screenshot of two side-by-side dropdown menus.

- The left menu is titled **Select Section(s)** and has a list of options: 'Training Website', 'About', 'Contribute', and 'Feedback'. 'Feedback' is currently selected.
- The right menu is titled **Select Document Type(s)** and has a list of options: 'Document', 'EmailIndex', 'Failed Import', and 'Feedback'. 'Feedback' is currently selected.

4. Now you may specify any additional selection criteria for the authors collection. The **Select content criteria** allows you to choose any of the elements that appear within the authors web pages and other documents, and you may select to match values within those fields. In this case select **CountryName** and select a comparison value of **IS EMPTY**. This will search for any Feedback items, which have a country name which is blank.



5. The author is now ready to generate the collection. Just click on the generate link and the author will see a document window pop up which has the documents which match the authors selection. The author can directly edit the documents form this list.

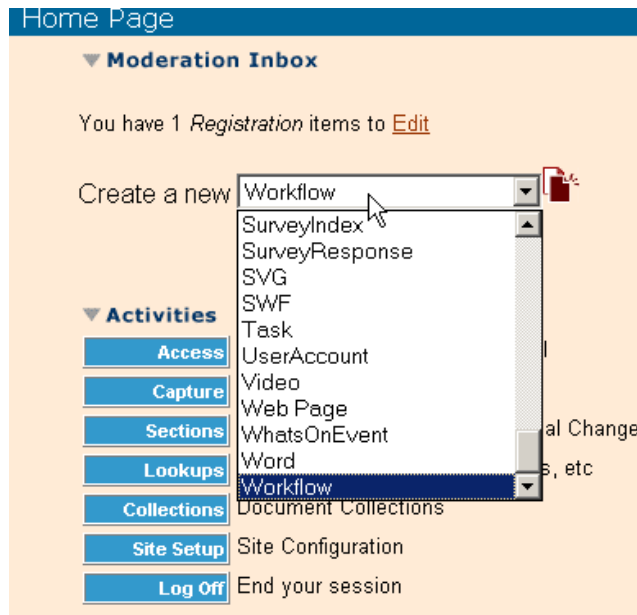


6. Click on **Collections** to return to the main list of Collections that are available. From the Collections list the author can return to this document list at any time. You may need to regenerate the list to pick up recent changes.

Creating a Workflow

The workflow feature allows the definition of data and event driven workflows. A workflow event triggers the transformation of the object from one document type to another, or from one published status to another. The process is linear in its simplest form, but can be extended using web services. The following process describes the creation of a simple linear workflow.

(1) From the Home Page, select Workflow and click the create button.



(2) The author will be prompted for the name and section for this workflow action. **The workflow name should be the name of the document type to which this workflow applies.**

 The screenshot shows a form for creating a workflow. It includes a 'Document Type' dropdown set to 'Workflow' and a 'Content Type' field set to 'Workflow'. Below this, a prompt says 'Please supply the details below:'. The form has three main sections: 'Workflow Title' with a text input field containing 'Scholarship Pending'; 'Language' with a dropdown set to 'English'; and 'Select Section' with a dropdown set to 'Study - Scholarships'. At the bottom left of the form is a red square button with a white arrow pointing up.

(3) The author may now specify the actions resulting from a PUBLISH and an UNPUBLISH action. The key action options are:

- what the next document status should be (default is "Published")
- what the next document type should be
- who should get an email alert on the publish avert occurring
- the publishing actions to be taken
 - No items checked – the normal publishing actions will occur:
 - Publish the document to the current publishing URL's
 - Change the Published status to "Published" if publishing was successful
 - None checked – no publishing actions will occur – the document status does not change.
 - Change document Type – checked – will change the document type to the type nominated above
 - Adjust publishing destinations – will change the publishing destinations to those specified in the proforma document for the *new* document type
 - Do not publish checked – the document *not* be published to its current Publishing destinations.