Pitney Bowes Business Insight

Customer
Communications
Management

DOC1® User's Guide

DOC1 Designer, DOC1 Generate

Version 5.5.1 Issue 1a

Contents

Preface

This guide covers the components, programs and features that are supplied as part of the DOC1 Designer and DOC1 Generate licensed products. It also provides detailed information about the DOC1 Work Center which acts as the repository and launch pad for DOC1 design and control objects.

It is important to note that not all features and objects discussed in the guide will be immediately available to all users. The way in which the DOC1 Work Center is presented to users may be customized by an Administrator to accommodate a user's level of experience or role within the DOC1 workflow. An Administrator may also choose to implement a security strategy that restricts access to particular projects, files or objects within the repository. Finally, the DOC1 license owned by your company may not allow access to all design features and will probably restrict the systems on which DOC1 Generate may run and the types of output datastream it can produce.

DOC1 Series 5 is a new generation of document composition tool and resource files from earlier versions of DOC1 are not automatically compatible with the editors and production environment of the new product.

Other than in regard to the repository features of the DOC1 Work Center, this guide deals exclusively with the editors and programs that form part of DOC1 Series 5. If you are working with legacy applications you should refer to the Designers Guide and Production Guide that relates to the appropriate level of resources.

Document structure

All new users should read "Getting started" on page 18.

Novice users who need to create or amend page layouts are recommended to refer to the section "Designing Page Layouts" on page 26 for information about working with the document editor.

Additional publication design topics including those that may be restricted by an Administrator are discussed in the section "Advanced Publication Design" on page 82.

For information about creating and maintaining the definitions of the input data to be used by DOC1 Generate refer to "Identifying Input Data" on page 236.

When you are ready to make a publication live refer to "Into Production" on page 270.

Administrators who need to manage the objects within the repository and perform other housekeeping should refer to “Administrative and Repository Tasks” on page 367.

Skills and education

While every effort is made to provide sufficient information in this and the other user manuals, we also recommend that users attend a DOC1 education course. Your DOC1 supplier will be pleased to advise on course availability.

Updates to this guide

This guide may be reissued in electronic format from time to time to include corrections or additions that have been made since the original hard copy publication. These changes will be indicated with a change bar in the margins. The latest versions of all DOC1 user guides can be downloaded from www.doc1supportnet.com.

Getting started

The DOC1 Work Center can be configured to support a wide range of site requirements and what you see when you first login to the system will depend on your user profile. Some profiles will assume you only want to work with publication designs and the Work Center will launch with the document editor. Other profiles will launch the Work Center Manager that provides views of the file repository and access to other types of resources.

Your user profile also determines what types of resources you are allowed to work with. If you have created a new repository when installing DOC1 (typically done by the first DOC1 user at a site) you will have Administrator rights by default and have access to all file types.

Workflow

DOC1 provides an integrated environment for designing and publishing personalized customer communications. Within the DOC1 Work Center the document editor allows you to combine page layouts and document logic with references to variable data typically emanating from your customer data bases. Specialized controls allow you to handle the repeating data – typically transaction records or similar – that is commonplace in these kind of communications.

In the production environment such variable data is provided to DOC1 in an input data file and before a publication can be put into production the type and structure of the input data must be defined within a data format. However, you do not need a data format to work with variable data in your designs. Data aliases are used to allow you to indirectly indicate the records and fields that will be available at production time. These are grouped into data dictionaries to allow reuse and control. Once a data format has been defined you can map the data fields to the actual structures in the input data.

Document designs and other design resources are stored in a **repository** that provides version control and multi-user access management where required. When working with design objects you can save them as *drafts* but before they can be used in production they must be *issued* which creates a new revision within the repository.

The Work Center also allows you to specify the settings that will control a publication in the production environment, i.e. information about the intended output device and the host system on which the job will run. At publication time, the document designs, data format, production information and the resources required to actually present the output from the publication are merged into a single file that is used by DOC1 Generate to manage production jobs.

DOC1 Generate produces one or more output datastreams in a format suitable for your target output devices.

Design basics

A **publication** is the DOC1 file that is actually put into production. It references one or more **document** files which contain the actual page layouts and associated logic. If required, you can use the same document in multiple publications.

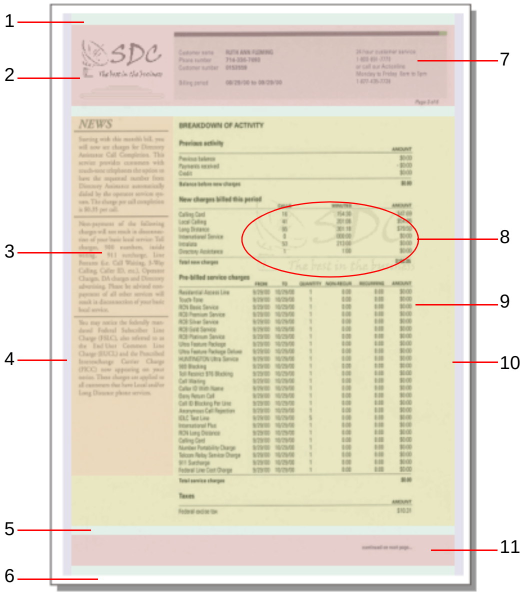
▶ **To start work on a new publication:** from the *File* menu select *New Publication*. A new document will be created automatically and the document editor window will be launched with this file loaded.

Page structure

When you start work on a new document you will typically be creating material for the body area of the pages to be created. The area available for the body is the remaining space once page margins, headers and footers, and any reserved areas have been taken into consideration. Content in the body area flows onto new pages as required.

Documents also have a **background** layer which you can use to provide a watermark or similar effect. You can also create a **reserved area** on the background to specify that part of the page is unavailable for the body area. Text and other content can be added to a reserved area as normal and it is possible to update the content dynamically if required.

Headers and footers can be used to specify content that is to be repeated at the top or bottom of each page. The content of headers and footers will often include system variables that provide page numbering functions and other information about the run-time environment.



1 Top margin

2 Header

3 Reserved area on the background layer

4 Left margin

5 Bottom margin

6 Unprintable area

7 Text box

8 Graphic on the background layer

9 Body

10 Right margin

11 Footer

You can create **text boxes** that act as a container for specific content. You can use a text box simply to highlight particular content or as a reserved area which text will flow around.

All **graphics** including generic shapes, text boxes, charts and references to external image files are positioned relative to their predecessor in the document flow by default. You can also use fixed or inline (within paragraph text) **anchor** positions.

Including data references

At the design stage DOC1 uses indirect references to data elements that will be available in the input data file to be used in the production environment. By using such **data fields** and **data records** you are able to design publications without needing to define the structure of the expected input data in advance. These aliases are mapped to actual elements of input data before the publication is put into production. It also means you can easily reassociate the aliases with new data structures if required.

You can create new data fields or records as you work or reference existing ones. Data aliases are grouped together into a **data dictionary** that, for example, represents all the data available to a particular type of application.

You can insert references to most data fields directly into paragraph text as required. However, when you need to reference elements within a **repeating data** structure – records or fields that appear iteratively such as transaction details – you will need to use a transaction table to control how they are positioned.

Tables and transaction tables

DOC1 uses tables for two distinct functions: a static table is used to align text and graphics in rows and cells or generally to help with positioning objects on a page (as with most word processing tools). A transaction table is used to control how repeating data is presented, i.e. how multiple iterations of the same data structure are ordered and how they are positioned in relation to one another.

The Table Wizard allows you to create the basic structure quickly and easily and controls for any type of table. Once created you can modify the table layout and settings as required.

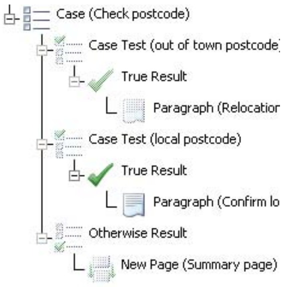
Document logic

When you are working with document designs you are actually doing two things:

- You are designing the look of the document pages, i.e. the text and graphics to be presented and their relative positions.
- You are specifying the sequence of objects and associated logic that will determine how the publication is processed in the production environment.

When you are working with simple designs you may often not need to be aware of the logic that is being created. It is simply a flow of paragraphs, tables and graphics across document pages.

However, when you need to work with more complex design issues, for example when using one of the functions provided, you will need to expose the document logic so that you can specify a processing sequence and provide parameters to objects.



Case (Check postcode)
Case Test (out of town postcode
True Result
Paragraph (Relocatior
Case Test (local postcode)
True Result
Paragraph (Confirm lo
Otherwise Result
New Page (Summary page)

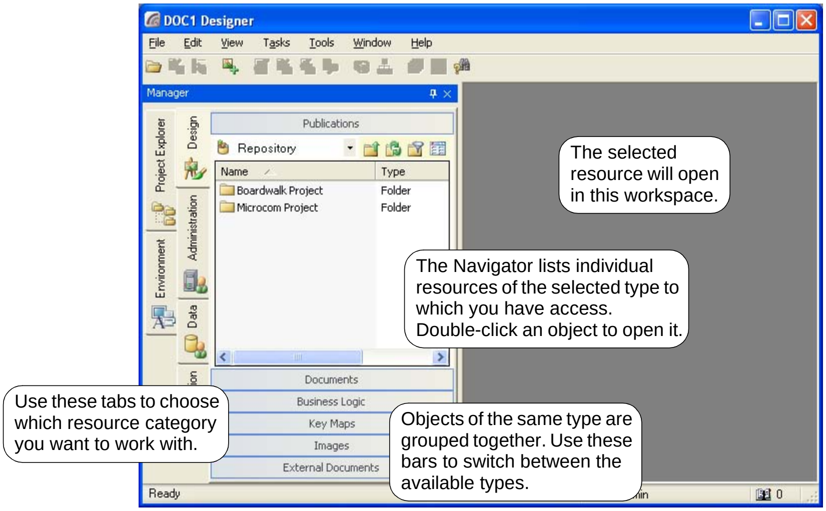
Document logic is displayed using the **Logic Map** view. When enabled, this will appear in the document editor alongside the page layout view when working with a publication design file.

You can actually decide to place paragraphs using simple conditions without exposing document logic. See “Conditional paragraphs” on page 44 for details.

▶ **To turn on the logic map view:** from the *View* menu select *Logic Map*.

Using the Work Center Manager

The Work Center Manager allows you to access and control all the DOC1 resources which you are allowed to use.



The selected resource will open in this workspace.

The Navigator lists individual resources of the selected type to which you have access. Double-click an object to open it.

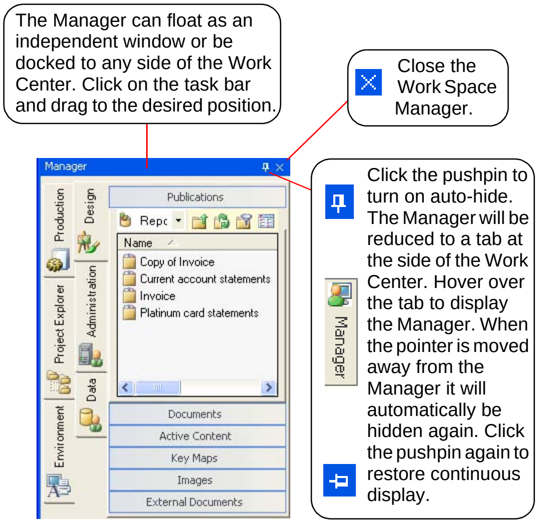
Use these tabs to choose which resource category you want to work with.

Objects of the same type are grouped together. Use these bars to switch between the available types.

▶ **To turn on the Work Center Manager:** from the *View* menu select *Manager/All*.

▶ **To show a resource category not currently displayed:** select the relevant category from the view menu.

The Manager appears as a separate dockable window within the Work Center and can be resized as required. You can also choose to use the auto-hide feature that means the Manager will only be shown when it is actively being used. This can be useful, for example, if you need more room when editing a document design.



The Manager can float as an independent window or be docked to any side of the Work Center. Click on the task bar and drag to the desired position.

Close the Work Space Manager.

Click the pushpin to turn on auto-hide. The Manager will be reduced to a tab at the side of the Work Center. Hover over the tab to display the Manager. When the pointer is moved away from the Manager it will automatically be hidden again. Click the pushpin again to restore continuous display.

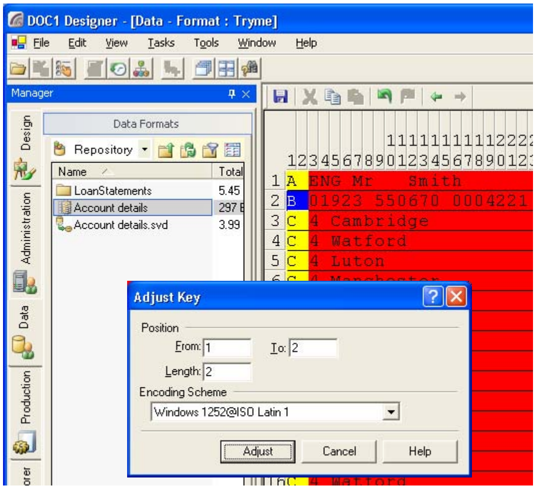
Input data and the data format

A data format defines the structure of the input data file that will be used with a DOC1 publication in the production environment. Any publication that needs to use input data must include a reference to a data format (as part of a *Host* settings object) before it can be put into production.

DOC1 supports three general types of input data:

- **Keyed Records** – a line data or 'flat' file where each line is an individual record. Record types are identified according to the contents of a consistent field within each line of data (the *key field*). Other fields within the records must be individually identified. You will need to supply a sample data file that acts as the template on which the record and field definitions are built.
- **Delimited** – a variation of keyed record data where field locations are automatically identifiable to DOC1 by the presence of one or more consistent delimiter characters between each field.
- **XML** – the structure of the data is automatically defined by the XML but note that the data types of individual fields must still be defined.
  Note that XML support is an optional extra and is only enabled to users who install both the DOC1 Work Center and Generate with the appropriate license keycodes.

Using the Data tab of the DOC1 Work Center you can import a sample of the intended input data and from this create a data format file. You can then use the data format editor to define the detailed structure and content of the data. You can also import a predefined data format from DOC1 Suite 4 and apply it to your sample data.



Name	Total
LoanStatements	5.45
Account details	297 E
Account details.svd	3.99

```
          1111111111222
 1234567890123456789012
1 A ENG Mr    Smith
2 B 01923 550670 0004221
3 C 4 Cambridge
4 C 4 Watford
5 C 4 Luton
6 C 4 Manchester
16 C 4 Watford
```

From: 1 To: 2
Length: 2
Encoding Scheme: Windows 1252@ISO Latin 1

Using the data format editor with keyed record sample data

In the document editor, references to input data elements are stored as part of a **data dictionary**. At publication time, a **data map** provides a cross reference between the data elements defined within a data dictionary and those in the assigned data format. If required you can export a data dictionary from a data format so that the document design process can be aware of the exact data elements that will be available.

Publishing and production

Publications are prepared for the production environment using a **Publish** task in the DOC1 Work Center. You can choose to produce a single HIP file that will control the job in the production environment and which also normally contains the resources required to actually print/present application output on the selected output devices. You can also however, choose to publish the resources and the design separately. The settings you specify when publishing can be saved in a **production job** object for reuse. Depending on the type of publishing task selected, before you can publish a publication design you will need to provide information about the printer/browsers for which output is to be created in one or more **output device** objects. You will also need to create a **host** object that provides operating system and deployment information regarding the intended production platform.

The publishing process also updates a data map to provide links to the data format for any unresolved data aliases in your publication designs.

At the completion of a publishing task a HIP file is created and is deployed to the location indicated in the selected host object. If the necessary connectivity is available, deployment can automatically copy the HIP to your host platform using FTP or APPC connections or you can deploy the file locally and manually copy to your production platform.

In the EngageOne Interactive environment, font and image resources are placed by the EngageOne publishing task within independent files that must be created and deployed in conjunction with requests from the EngageOne server. For more information about EngageOne see “Creating EngageOne templates” on page 201.

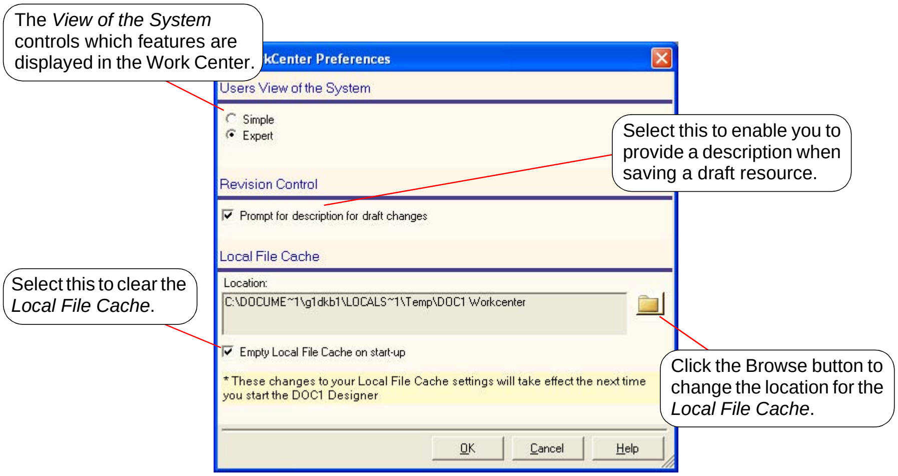
In the production environment DOC1 **Generate** is the batch program that processes jobs on your chosen host system. You will need to specify the HIP file that controls the publication as a parameter when starting the program. Generate merges input data with your publication designs and produces output datastreams ready for printing or presenting on your intended output devices.

Work Center Preferences

The DOC1 Work Center provides a number of general options that enable you to customize your system. These Work Center preferences enable you to do the following:

- select whether to display all the functionality of the Work Center or to hide some of the more advanced features from users
- provide a description of resources when saving a draft version of them. These descriptions are displayed when you go to revert to a previous version or display the history of the resource. The option to prompt for a description can be overridden for individual resources, see “Revision control” on page 378 for more details.
- change settings for the *Local File Cache*. This is where files in the DOC1 Design Repository are placed when they are in use. You can choose to change the location for the file cache, as well as have it cleared each time you restart the Work Center.

▶ **To access the Work Center preferences:** from the *Tools* menu select *Work Center Preferences* and the dialog box is displayed.



kCenter Preferences

Users View of the System

○ Simple
● Expert

Revision Control

☑ Prompt for description for draft changes

Local File Cache

Location:
C:\DOCUME~1\g1dkb1\LOCALS~1\Temp\DOC1 Workcenter

☑ Empty Local File Cache on start-up

* These changes to your Local File Cache settings will take effect the next time you start the DOC1 Designer

The *View of the System* controls which features are displayed in the Work Center.

Select this to enable you to provide a description when saving a draft resource.

Select this to clear the *Local File Cache*.

Click the Browse button to change the location for the *Local File Cache*.

Errors and information

In the DOC1 Work Center errors, warnings and information messages from any activity are shown in the Event Log.

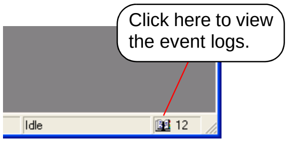
Messages have a unique message ID associated with them which you should quote if you need to contact your DOC1 supplier when reporting a problem. A severity tag is also displayed with each message:

ℹ **Information** only. No user action required.

⚠ **Warning**. An abnormal event that may require user action.

✖ **Error**. An abnormal event that cannot be corrected without user intervention.

▶ **To display the event log:** click on the logs icon in the status bar.



Idle | 12

Click here to view the event logs.

In the production environment messages are written to the standard output device for the platform.

Designing Page Layouts

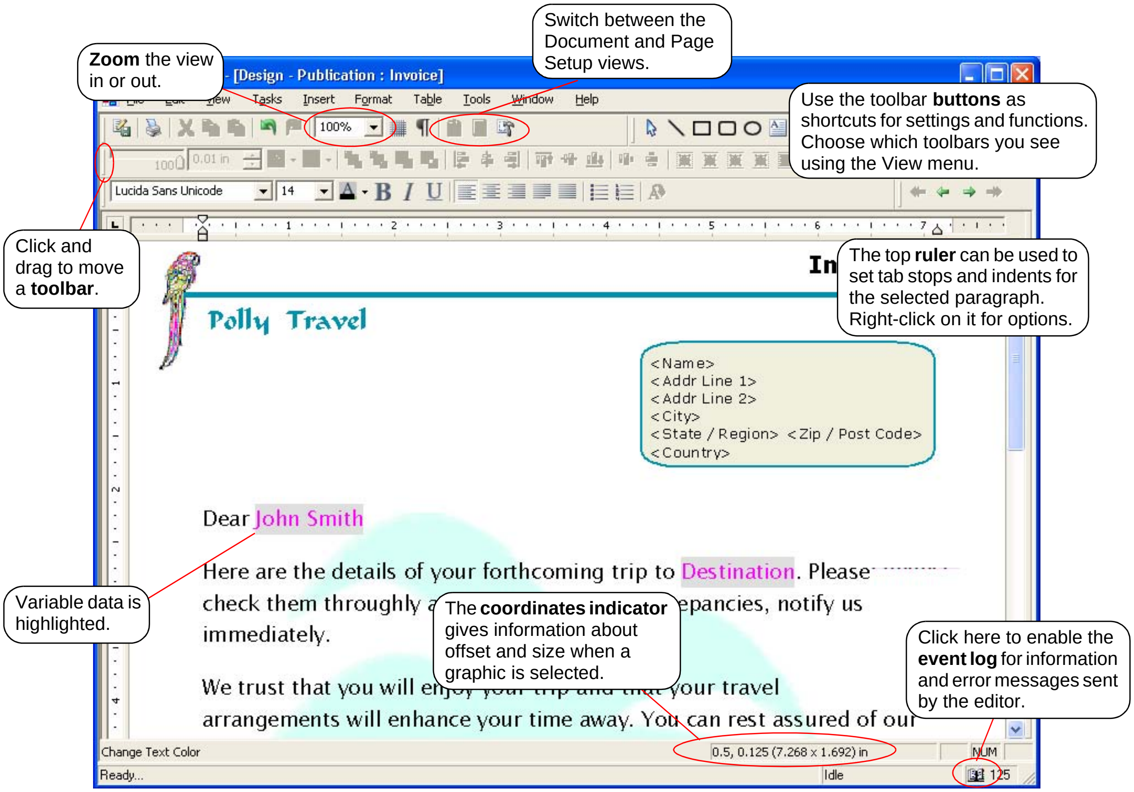
About the document editor

> As well as the main page work space, the logic map view may also be enabled when the editor is launched. Refer to “Using the logic map” on page 92 for details.

The document editor is where you design actual page layouts. When you create a new publication the editor will be invoked with a blank page ready for your input. Depending on your user profile this may also be the view you see by default when the DOC1 Work Center is initiated.

▶ **To create a new publication:** in *File* menu select *New Publication*.

By default, you will be working with the document view where content intended for the body area of the document is created. You may also want to work with the page setup view which allows you to create content for headers and footers and the background layer. Both views share the same editing features. See “Page layout” on page 60 for details of these views and the page areas they allow you to define.


[Design - Publication : Invoice]
File Edit View Tasks Insert Format Table Tools Window Help
100%
100
0.01 in
Lucida Sans Unicode
14
B I U
1 2 3 4 5 6 7
In
Polly Travel
Change Text Color
0.5, 0.125 (7.268 x 1.692) in
NUM
Ready...
Idle
125

<Name>
<Addr Line 1>
<Addr Line 2>
<City>
<State / Region> <Zip / Post Code>
<Country>

Dear John Smith

Here are the details of your forthcoming trip to Destination. Please check them throughly a … epancies, notify us immediately.

We trust that you will en … your travel arrangements will enhance your time away. You can rest assured of our

Zoom the view in or out.

Switch between the Document and Page Setup views.

Use the toolbar **buttons** as shortcuts for settings and functions. Choose which toolbars you see using the View menu.

Click and drag to move a **toolbar**.

The top **ruler** can be used to set tab stops and indents for the selected paragraph. Right-click on it for options.

Variable data is highlighted.

The **coordinates indicator** gives information about offset and size when a graphic is selected.

Click here to enable the **event log** for information and error messages sent by the editor.

Editing features

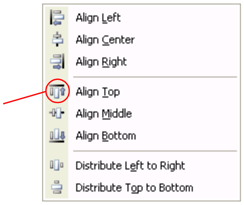
The editor provides standard editing features that enable you to create documents. The menus and toolbars give you access to all the editing options. The layout of these can be customized to some extent.

Text and graphics can be copied and moved by selecting it and using the clipboard as a temporary store for it. If you make a mistake when editing, you can retrace your steps using the undo and redo options.

Menus and toolbars

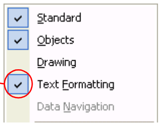
The menu gives you access to most of the editor functions and commands. You will also be able to access many of these using the toolbar buttons or shortcut keys, both of which will be shown on the menu.

Also available as a toolbar button



Buttons are grouped together into a range of separate toolbars that cater for a particular group of editor functions. You can hide toolbars that you don't need to work with to simplify the layout, or to make more space for the editor window. For instance, if your design does not include any shapes you may want to hide the drawing toolbar. You can also move the toolbars in relation to one another to suit your own style of working.

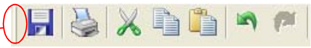
Tick to select option



▶ **To hide or display a toolbar:** on the *View* menu click *Toolbars* and then clear or select the relevant toolbar check box.

▶ **To move a toolbar:** drag its docking bar to a new position.

Docking bar



Measuring and positioning aids

You can customize the unit of measure (UOM) used by the editor to suit your requirements. The UOM specified is used in rulers and coordinate displays.

The preferred UOM is also used as the default in dialogs that require a measurement although you can override this as required. You can specify a different UOM simply by adding one of the following abbreviations after the number.

UOM	Specify	Example
inches	in	2.1 in
centimeters	cm	1.2 cm
millimeters	mm	32 mm
points	pt	12 pt
picas	pi	2 pi

The on-screen **rulers** help you with positioning objects accurately. The top ruler is also used when setting tab stops and indents for paragraph text. You can turn off the ruler display if required by using the *View/Ruler* option.

ⓘ FOR INFORMATION ABOUT SETTING TAB STOPS AND INDENTS USING THE RULER SEE “POSITIONING TEXT” ON PAGE 37.

You can also use the **grid** to help with placement. You can set the grid size to your preference and optionally have graphics snap to grid intersections when you are creating or moving them.

▶ **To set all grid, ruler and UOM preferences:** from the *Tools* menu select *Options* and apply settings as required.

▶ **To override ruler settings and UOM:** right-click on the ruler and select from the drop down list.

▶ **To turn the grid on and off:** click on the *Toggle Grid* button ▦
– or –
on the *View* menu click *Grid*.

Selecting text and graphics

When you want to move, delete, or overwrite design elements you must first select what you want to work with. You can do this using the mouse or the keyboard.

▶ **To select text and graphics using the mouse:** either drag the pointer across what you want to select or use one of the following shortcuts:

Anything:	drag across it from beginning to end.
A word:	double-click in it.
Start a selection:	click at start of selection.
End a selection:	hold down SHIFT and click at end of selection.

▶ **To select paragraph text using the keyboard:** use one of the following shortcuts:

A character to the right:	SHIFT + →
A character to the left:	SHIFT + ←
A line before:	SHIFT + ↑
A line after:	SHIFT + ↓
To end of line:	SHIFT + END
To beginning of line:	SHIFT + HOME
All	CTRL + A or on *Edit* menu click *Select All*

▶ **To cancel a selection:** click anywhere else on the page or press any arrow key.

▶ **To delete selected elements:** use the *Delete* key on the keyboard or click *Clear* from the Edit menu.

Using the clipboard

You can copy, cut and paste design elements using the system clipboard. This is an easy way of transferring content both within the DOC1 editor and to and from other applications.

▶ **To use the clipboard:** do one of the following:

Copy selected text and graphics onto the clipboard:	on *Edit* menu click *Copy* or press ⎘ or CTRL+C
Move (i.e. copy and delete) selected text and graphics onto the clipboard:	on *Edit* menu click *Cut* or press ✂ or CTRL+X
Paste contents of the clipboard into the document:	on *Edit* menu click *Paste* or press 📋 or CTRL+V
Paste contents of the clipboard, such as content from a spreadsheet or PDF into the document:	on *Edit* menu click *Paste Special*. If applicable, select the format you want to use to paste the contents and click *OK*.

Undo and redo

If you create, edit or format a design element by mistake, you can reverse the action by using the *Undo* command. Similarly, if you undo an action by mistake, you can use the *Redo* command to switch back again. You can undo actions progressively, each time restoring to the state before the previous change was made, up to a limit of 10 previous actions.

▶ **To undo an action:** click the *Undo* button
– or –
on the *Edit* menu click *Undo*. Note that the action that will be canceled is shown in the *Undo* command.

▶ **To reverse an undo action:** click the *Redo* button
– or –
on the *Edit* menu click *Redo*. Note that the action that will be reversed is shown in the *Undo* command.

Find and replace

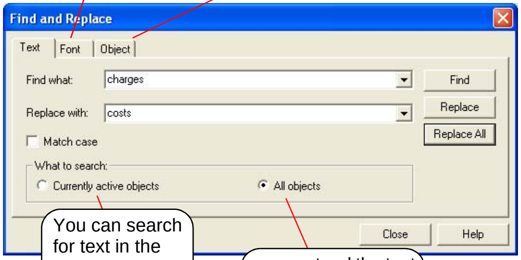
You can search and make changes to text, fonts, character styles and value objects in your design documents by using the find and replace feature. You can replace specific instances of text and value objects or globally change every occurrence within the document. For fonts and styles you can replace every occurrence in both Design view and Page Setup view.

NOTE THAT IN A PREVIEW FOR ENGAGEONE YOU CAN ONLY FIND AND REPLACE TEXT IN AN EDITABLE TEXT SECTION. ALSO THE *WHAT TO SEARCH* OPTIONS ARE NOT APPLICABLE FOR ENGAGEONE. FOR INFORMATION ON SEARCHING THE LOGIC MAP SEE “USING THE LOGIC MAP” ON PAGE 92. FOR DETAILS ON WORKING WITH VALUES SEE “ADDING VALUES TO OBJECTS” ON PAGE 104.

▶ **To find and replace:** from the *Edit* menu select *Find and Replace*. Select the relevant tab to search for *text*, *fonts* or *objects*.

The *Font* tab is used to find and replace the font, font style, size, color and language.

The *Object* tab is used to find and replace values e.g. data fields and variables.



Find what: charges

Replace with: costs

□ Match case

What to search:
○ Currently active objects ● All objects

You can search for text in the editor only …

… or extend the text search to the logic map.

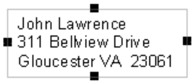
Refresh

The refresh option is used to refresh the view of the document being edited in the editor.

▶ **To refresh the editor view:** select *View/Refresh*
– or –
press F5.

Boundary indicators

You can optionally choose to show the outline of layout areas such as text boxes, address blocks, charts and so on. Boundaries are indicated by a grey outline around the layout area.



John Lawrence
311 Bellview Drive
Gloucester VA 23061

Select the object by clicking anywhere in the layout area or on its border.

▶ **To switch boundaries on and off:** click the □ toolbar button
– or –
from the *View* menu choose *Show Boundaries.*

Note that by default boundaries are switched on.

Creating paragraphs

Entering and formatting text

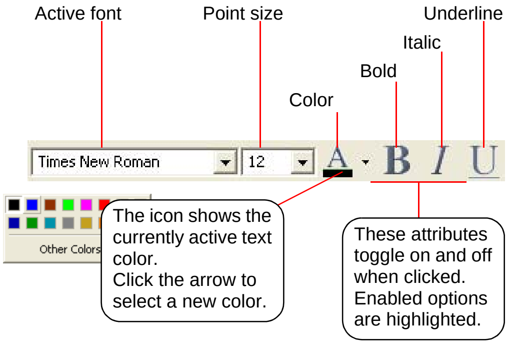
Paragraphs are created simply by typing with the insertion point positioned in the body area of the document view or in a control area such as a text box or header and footer space. A new paragraph is automatically created when you press the return key. You can also create new paragraphs by double-clicking in any open space below the previous paragraph in the document view.

Text will automatically wrap onto new lines when the available width is exceeded, using hyphenation when specified. Pressing the shift and return keys together forces a new line without creating a new paragraph. In the main body area text also flows onto a new page when the current page is full.

When you start typing, the text is created using the active format – the font being used, its style, color and so on. You can change the format at any time or amend the format of existing text as required. Note that when you move the insertion point into existing text, the current format will be picked up from the text to the right of the insertion point. You can, of course, change the format before typing.

Text attributes are normally specified using the text formatting toolbar. Note that you can specify a global default font – see “Default editor font” on page 34.

Changes to the formatting of text will be applied to any highlighted text, or, if none is highlighted then to any new text that is entered at the insertion point.



The icon shows the currently active text color. Click the arrow to select a new color.

These attributes toggle on and off when clicked. Enabled options are highlighted.

▶ **To show the text formatting toolbar:** select *View/Toolbars/Text Formatting* from the menu.

▶ **To adjust the formatting of text:** highlight the text to be changed and either use the text formatting toolbar options
– or –
on the *Format* menu click *Font*. This will invoke the *Font* dialog box where you can format the selected text.

Right to left text entry

The default direction of paragraph text is determined by the active Windows input language (see “Changing the input language” on page 34). This is generally left to right, but for languages such as Arabic, would be right to left.

When the direction is right to left, text is entered using the keyboard in the same manner as if typing left to right text. Paragraphs can include bidirectional text and you must ensure that the direction of the paragraph is correct.

▶ **To set the direction of a paragraph to right to left:** invoke the corresponding *Paragraph* dialog box. Under *Presentation* set the *Alignment* as required and the *Direction* to *Right to Left*.
– or –
use the *Right-to-left* ¶◂ button on the Text formatting toolbar.

Default editor font

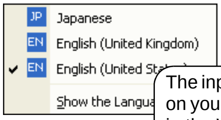
The editor’s default font is used for text entered in publications and presentation objects, such as address blocks and text boxes. The default font is set using the *Options* dialog box and enables you to choose the font, point size etc.

You can use the formatting toolbar to change font attributes without affecting the default font. Changes made to the default font do not affect the text in existing publications and presentation objects.

▶ **To change the default font:** select *Tools/Options*. Click the *Edit* tab and choose the required font.

Changing the input language

The input language controls how characters will be mapped to the font used within an output datastream – i.e. the code page to be used. Normally the intended language of text is automatically assumed from the active Windows input language when characters are entered. In some exceptional circumstances this may not be the case and you will need to set the intended language for specific text. You can use the Unicode Language option for complex scripts and languages that do not map to a standard Windows character set.



The input languages installed on your system are displayed in the Windows taskbar.

▶ **To set the language of existing text:** highlight the text you want to change and select *Tools/Set language*. Select a supported language and click *OK*.

ⓘ WHEN CREATING YOUR DOCUMENT ENSURE THAT YOU HAVE THE CORRECT INPUT LANGUAGE SET OR YOU MAY BE UNABLE TO SELECT SOME LANGUAGE SPECIFIC FONTS IN THE EDITOR.

Changing the language mapping

Normally the default encoding for an input language is sufficient. However, there may be situations when you may want to change that encoding, for instance to view a field from a data format that uses a different encoding scheme to the one used by the publication. This option is only used for viewing content in the DOC1 editor and is done to prevent the Windows operating system from automatically substituting fonts in a publication.

An additional option is available for changing the language mapping when text is entered in the DOC1 editor using a Windows Input Method Editor (IME). If this option is selected when characters are entered into a document using the IME, the characters displayed in the document will use the mappings that correspond to the custom code page in DOC1 and not the character set associated with the language and font of the IME.

> ⓘ Changing the encoding for one language, may impact a number of other languages that share the same encoding. These languages are displayed in the *Affected languages* list.

▶ **To change the language mapping:**

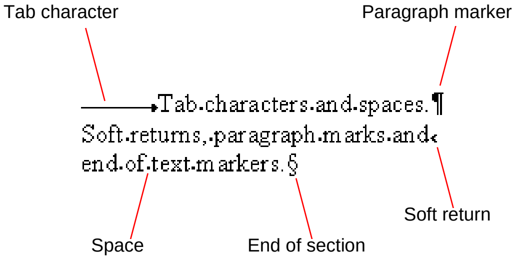
1. Select *Tools/Options* and select the *Language/encoding mapping* tab.
2. Select the language you want to change and click *Modify*.
3. Select either a *System* or *Custom* code page you want to use from the *Encoding* list and click OK.
4. The new mapping is now displayed in the *Options* dialog box. Click *OK*.

▶ **To use custom code pages for keyboard entry:** select *Tools/Options* and select the *Language/encoding mapping* tab and click the *Use custom code pages for keyboard entry* option.

▶ **To reset the language mapping:** select *Tools/Options* and select the *Language/encoding mapping* tab and click *Reset All*.

Formatting marks

Formatting marks are non-printing characters that indicate controls within paragraphs such as end markers, spaces, tab characters and so on. Displaying them shows you exactly what you have typed and can help to highlight layout problems, for example, where you have typed several spaces rather than a tab character. Hiding them shows you more how the finished page might look and also removes empty paragraph objects from the logic map. Even if formatting marks are showing, they will not be in the final output.



▶ **To show/hide formatting marks:** click the *Formatting Marks* ¶ button or select *View/Show all Paragraphs* to switch them on or off.

Hyphenation

Text in most Latin based languages can be automatically hyphenated by DOC1 if required. This is initially set for the entire document but can be overridden as necessary for specific paragraphs.

Note that text in right-to-left paragraphs can not be hyphenated.

▶ **To set hyphenation for a document:** in either the Publication view or the Document view double-click on the relevant document. In the *Document* dialog box select or clear the *Enable hyphenation* option.

▶ **To turn off hyphenation for a specific paragraph:** select the relevant paragraph and on the *Format* menu click *Paragraph*. In the *Presentation* section of the *Paragraph* dialog box select the *Don't hyphenate* option.

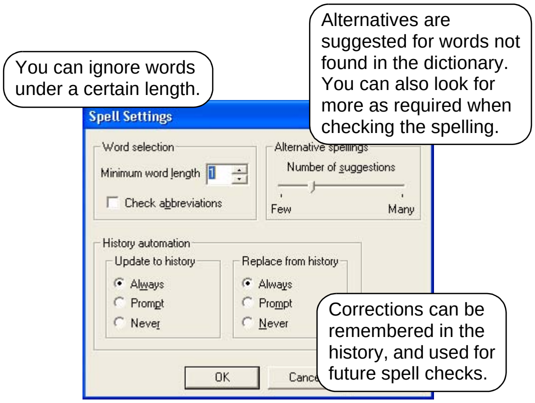
Spell checking

Spell checking is performed using the dictionary associated with the current DOC1 locale. Each locale has a default dictionary associated with it. You can change the locale's dictionary, but you must ensure that the language corresponds to the locale for the spell checker to be accurate.

For more information on locales, see "Localization" on page 183.

If text is selected, then only this will be checked, otherwise, all text will be checked. You can customize the way the spell checker works by changing the spell settings – for instance, whether to check abbreviations, or whether to remember previous changes. Note however, that spelling settings are not applicable for EngageOne templates when using the preview feature. See "Previewing an EngageOne template" on page 216.

▶ **To run the spell checker:** on the *Tools* menu select *Spell check*.



You can ignore words under a certain length.

Alternatives are suggested for words not found in the dictionary. You can also look for more as required when checking the spelling.

Corrections can be remembered in the history, and used for future spell checks.

Spell Settings

Word selection

Minimum word length: 1

□ Check abbreviations

Alternative spellings

Number of suggestions: Few – Many

History automation

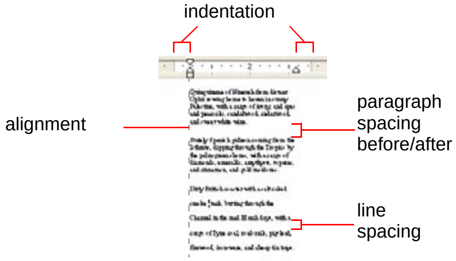
Update to history: ● Always ○ Prompt ○ Never

Replace from history: ● Always ○ Prompt ○ Never

▶ **To change the spell settings:** on the *Tools* menu select *Spell settings* and change the settings as required.

Positioning text

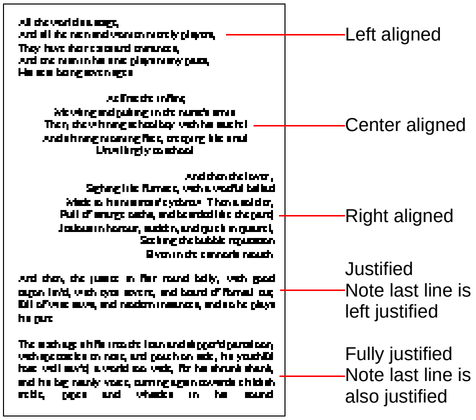
Paragraphs are positioned in relation to one another and the edges of the working area; for example, the margins of the body area or the sides of a text box or table cell. You can customize how they are aligned (or justified) between the left and right edges, any indentation to be used from those edges, the line spacing within individual paragraphs and the gap required between adjacent paragraphs.



Tab stops can also be used to align text within a paragraph.

Paragraph alignment

Paragraphs are aligned between their left and right indentations – shown as ⌂ on the ruler. Alignment is applied to the whole paragraph, so if you want a different layout for certain lines, you must create a separate paragraph for them.


Left aligned
Center aligned
Right aligned
Justified
Note last line is left justified
Fully justified
Note last line is also justified

▶ **To align a paragraph:** use one of the following buttons on the Text Formatting toolbar:

Left

Centered

Right

Justified

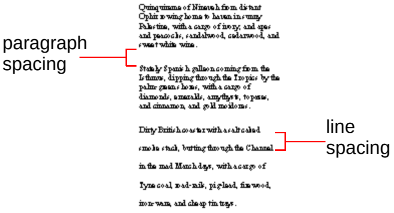
Fully justified

– or –

on the *Format* menu click *Paragraph*. In the *Presentation* section of the *Paragraph* dialog box set the required *Alignment* option.

Spacing

You can specify both space before and after a paragraph, and space between lines within a paragraph.


Quinquireme of Nineveh from distant Ophir rowing home to haven in sunny Palestine, with a cargo of ivory, and apes and peacocks, sandalwood, cedarwood, and sweet white wine.

Stately Spanish galleon coming from the Isthmus, dipping through the Tropics by the palm-green shores, with a cargo of diamonds, emeralds, amythysts, topazes, and cinnamon, and gold moidores.

Dirty British coaster with a salt-caked
smoke stack, butting through the Channel
in the mad March days, with a cargo of
Tyne coal, road-rails, pig-lead, firewood,
iron-ware, and cheap tin trays.

For paragraphs you can specify both the amount of space to be inserted after the preceding paragraph, and the amount of space before the following paragraph. If consecutive paragraphs have conflicting 'before' and 'after' values then the greater value will be used – for example if two paragraphs have the space above set to 0.1″ and the space after set to 0.2″, then the paragraph spacing will be 0.2″.

Line spacing determines the space between the lines within a paragraph.

Single: represents the standard amount of line spacing for the largest font in the paragraph. *1.5 lines* and *Double* are increments of this default.

Multiple: as for *Single*, but uses the value specified as a multiple of the line, e.g. 1.2 will increase the line spacing by 20%; 0.8 will decrease it by 20%.

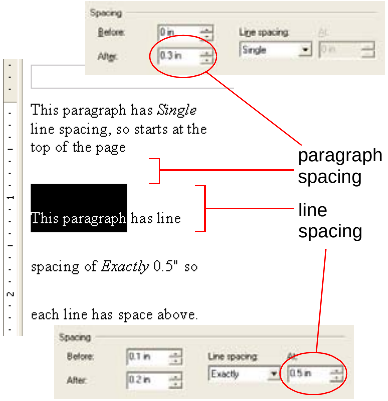
At least: lines will be spaced by at least the specified amount, but will be more as required by the content of each line.

Exactly: lines are spaced using the specified value, regardless of the size of the content of a line.

> ⓘ NOTE THAT ONLY *EXACTLY* WILL SPACE <u>ALL</u> THE LINES IN THE PARAGRAPH THE SAME. ALL OTHER SETTINGS CAN PRODUCE DIFFERENT LINE SPACINGS WITHIN A PARAGRAPH DEPENDING ON THE CONTENT OF EACH INDIVIDUAL LINE, FOR EXAMPLE IF THERE IS AN INLINE GRAPHIC.

Line spacing is measured from the bottom of the line upwards, so if you specify a value in *At* that is much larger than the font, the additional space is above the line. This can affect the gap between paragraphs as the paragraph spacing is applied in addition to line spacing.

Paragraph and line spacing should be used in preference to inserting blank paragraphs between text, as it gives more control over the overall layout on the page. For instance, if a paragraph doesn't quite fit you may be able to reduce the paragraph spacing and the line spacing so that it does – say by using *Multiple* to reduce the line spacing by a small amount, e.g. 0.7 of a line.


Spacing
Before: 0 in
After: 0.3 in
Line spacing: Single
At: 0 in

This paragraph has *Single* line spacing, so starts at the top of the page

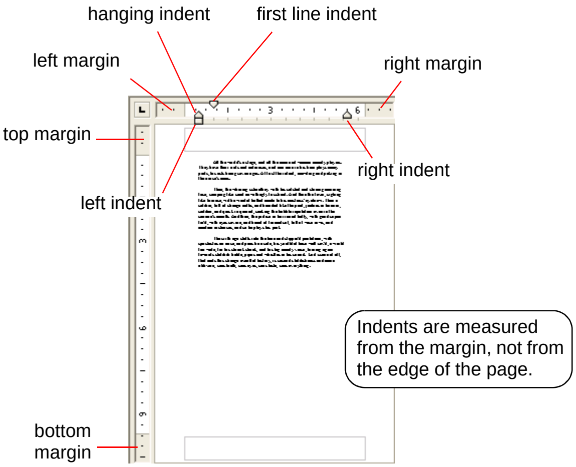
This paragraph has line spacing of *Exactly* 0.5" so each line has space above.

Spacing
Before: 0.1 in
After: 0.2 in
Line spacing: Exactly
At: 0.5 in

▶ **To adjust paragraph or line spacing:** on the *Format* menu click *Paragraph*. In the *Paragraph* section of the *Paragraph* dialog box set the required *Spacing* options.

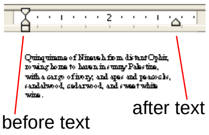
Indentation

Paragraphs are, by default, positioned at the left and right edges of the working area (for the body area, these are set by the page margins). However, you can specify a gap between these edges and the paragraph by applying indentation.

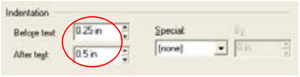


Indents are measured from the margin, not from the edge of the page.

▶ To indent a paragraph: move the left and right indent markers on the ruler to the required positions
– or –
on the *Format* menu click *Paragraph*. In the *Indentation* section of the *Paragraph* dialog box set the *Before text* and *After text* indentation.



Quinquireme of Nineveh from distant Ophir, rowing home to haven in sunny Palestine, with a cargo of ivory, and apes and peacocks, sandalwood, cedarwood, and sweet white wine.



Indentation

Before text: 0.25 in

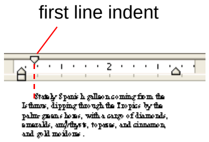
After text: 0.5 in

Special: (none)

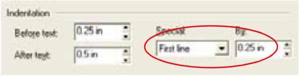
By:

Indenting the first line

You can indent the first line of a paragraph in relation to other lines it contains.



Stately Spanish galleon coming from the Isthmus, dipping through the Tropics by the palm-green shores, with a cargo of diamonds, emeralds, amethysts, topazes, and cinnamon, and gold moidores.



Indentation

Before text: 0.25 in

After text: 0.5 in

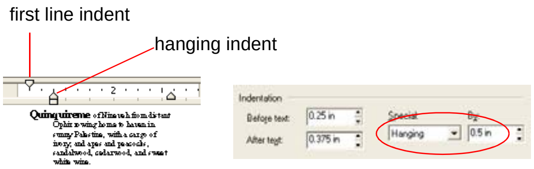
Special: First line

By: 0.25 in

▶ To indent the first line: move the first line indent marker on the ruler to the required position
– or –
on the *Format* menu click *Paragraph* and select *First Line* from the *Special* indentation drop-down list. Enter the amount to indent in the *By* field.

Hanging indent

A hanging indent is where all lines except the first one are indented. The hanging indent is measured from the left, or first line indent.



Quinquireme of Nineveh from distant Ophir rowing home to haven in sunny Palestine, with a cargo of ivory, and apes and peacocks, sandalwood, cedarwood, and sweet white wine.

Indentation

Before text: 0.25 in

After text: 0.375 in

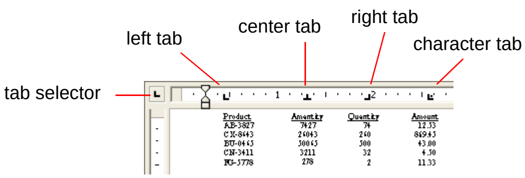
Special: Hanging

By: 0.5 in

▶ **To create a hanging indent:** move the hanging indent marker on the ruler to the required position
– or –
on the *Format* menu click *Paragraph*. Select *Hanging* from the *Special* indentation drop-down list. Enter the amount to indent in the *By* field.

Tab stops

Any number of tab stops can be specified in a paragraph. They can be left, right, centered, or aligned on a character. A character tab will align on the first occurrence of the specified character. If this is not in the text, then a tab the same as the paragraph justification will be used.



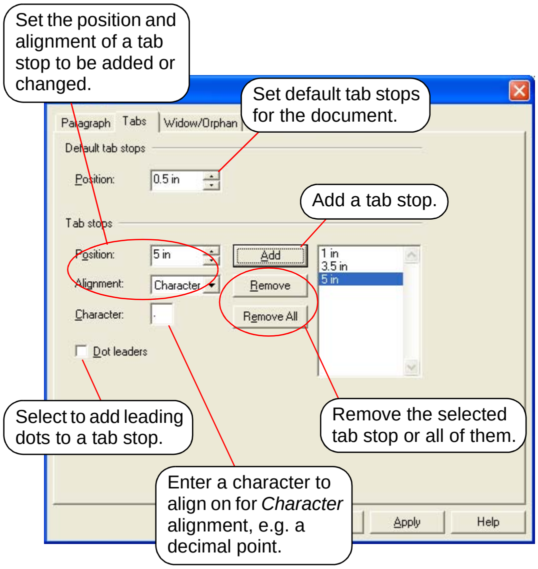
Product	Amentity	Quantity	Amount
AB-3827	7427	74	12.53
CX-8643	26043	260	869.45
BU-0465	50065	500	43.00
CN-3411	3211	32	4.50
NG-5778	278	2	11.33

If there are no tab stops specifically set in a paragraph, then, by default, tab stops will be set at regular intervals on the ruler. These will be the same as the paragraph alignment, e.g. right tabs in a right aligned paragraph.

Tab selector	Tab stop
└	left
┴	centered
┘	right
╘	character

▶ **To insert a tab stop:** click the tab selector on the ruler until the required tab stop is shown. Then click on the ruler at the position where you want to place the tab stop. The tab stop will be inserted on the ruler
– or –
on the *Format* menu click *Tabs*. The *Tabs* page of the *Paragraph* dialog box will be displayed. Specify a tab stop and click *Add*.

▶ **To move a tab stop:** either select the tab stop on the ruler and drag it to its new position
– or –
on the *Format* menu click *Tabs*, select the relevant tab stop from the list and enter the new position under *Tab Stops/ Position*.



Default tab stops
Position: 0.5 in

Tab stops
Position: 5 in
Alignment: Character
Character: .
☐ Dot leaders

1 in
3.5 in
5 in

Set the position and alignment of a tab stop to be added or changed.

Set default tab stops for the document.

Add a tab stop.

Select to add leading dots to a tab stop.

Remove the selected tab stop or all of them.

Enter a character to align on for *Character* alignment, e.g. a decimal point.

▶ **To modify an existing tab stop:** on the *Format* menu click *Tabs*, select the tab stop from the list and change its *Alignment*, *Character* and *Dot leaders* as required.

▶ **To remove a tab stop:** either select the tab stop on the ruler and drag it off the ruler
– or –
on the *Format* menu click *Tabs*, select the tab stop from the list and then click *Remove*, or click *Remove All* to delete all tab stops.

▶ **To remove all tab stops:** on the *Format* menu click *Tabs* and then click *Remove All*.

▶ **To set default tab stops:** on the *Format* menu click *Tabs*. Enter the interval between tab stops under *Default Tab Stops*.

Dot leaders

Dot leaders can be used between tab stops to help readability. When a leader is applied to a tab, a row of dots is inserted between the end of preceding text and the current tab stop. This is typical of table of contents and similar styles.

▶ **To insert a dot leader:** on the *Format* menu click *Tabs*, select the tab stop from the list and tick *Dot leaders*.

Rotating text

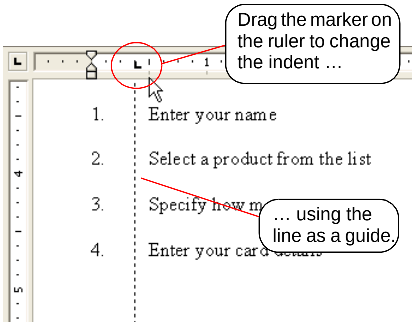
Text can be rotated only when it is in a text box, see “Rotating text” on page 54 for details.

Bullets and numbering

Bullets or numbering can be applied to paragraphs if you want to emphasize them or to create lists. When they are applied the paragraph indentation will automatically be changed to suit. Numbering can be reset at any time back to '1' and you can add a character to follow the number, for example, a dot. You can specify a different font (or font attribute) for the bullet or numbering character. By default, bullets use the '•' character, but you can change this, if required, to a different character.

▶ **To set bullets or numbering:** use the *Bullet* or *Numbering* toolbar buttons
– or –
on the *Format* menu click *Paragraph* and on the *Numbering/Bullets* page select the style that you want.

A hanging indent is automatically applied with a tab between the bullet/number and the text.



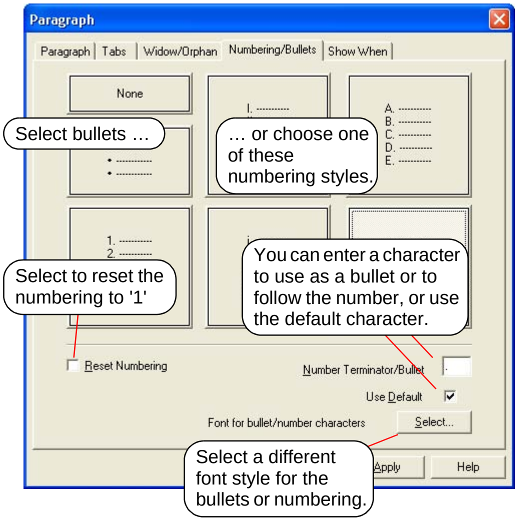
1. Enter your name
2. Select a product from the list
3. Specify how m
4. Enter your card details

Drag the marker on the ruler to change the indent …

… using the line as a guide.

▶ **To format bullets and numbering:** on the *Format* menu click *Paragraph* and on the *Numbering/Bullets* page select formatting as required.

▶ **To turn off bullets or numbering:** de-select the appropriate toolbar button.
– or –
on the *Format* menu click *Paragraph* and on the *Numbering/Bullets* page select *None*.



Paragraph
Paragraph | Tabs | Widow/Orphan | Numbering/Bullets | Show When
None
I. ----------
A. ---------- B. ---------- C. ---------- D. ---------- E. ----------
1. ---------- 2. ----------
☐ Reset Numbering
Number Terminator/Bullet
Use Default ☑
Font for bullet/number characters
Select...
Apply
Help

Select bullets …

… or choose one of these numbering styles.

Select to reset the numbering to '1'

You can enter a character to use as a bullet or to follow the number, or use the default character.

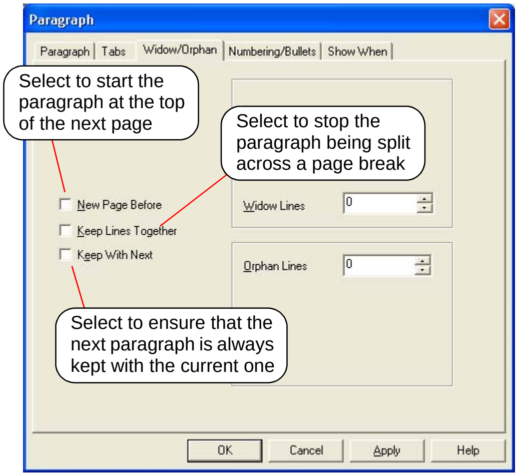
Select a different font style for the bullets or numbering.

Pagination

There are various options you can use that affect how a paragraph is positioned and the text flow. You can force the paragraph to start at the top of a page, keep a minimum number of lines together if a page break splits a paragraph (widows and orphans), or never allow a page break to split a paragraph so that the entire paragraph is kept together.

You can also specify that a paragraph is always kept with the following paragraph.

▶ **To control pagination:** on the *Format* menu click *Paragraph* and on the *Widow/Orphan* tab select the options required.



Paragraph

Paragraph | Tabs | Widow/Orphan | Numbering/Bullets | Show When

☐ New Page Before

☐ Keep Lines Together

☐ Keep With Next

Widow Lines 0

Orphan Lines 0

OK | Cancel | Apply | Help

Select to start the paragraph at the top of the next page

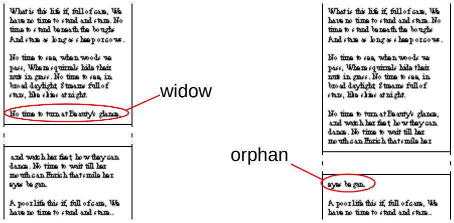
Select to stop the paragraph being split across a page break

Select to ensure that the next paragraph is always kept with the current one

Widows and orphans

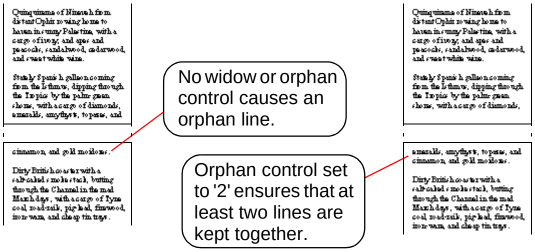
Widows and orphans occur when a paragraph is split across a page break, with the bulk of the paragraph on one page and the remainder on the other.

A widow is a few lines at the beginning of a paragraph that are separated from the rest of the paragraph by a page break. An orphan is a few lines at the end of a paragraph that are separated from the rest of the paragraph by a page break.



Specify the number of lines that you want kept together at the bottom or top of a page.

▶ **To specify widows and orphans:** on the *Format* menu option click *Paragraph* and on the *Widow/Orphan* tab enter the number of *Widow Lines* or *Orphan Lines* required.

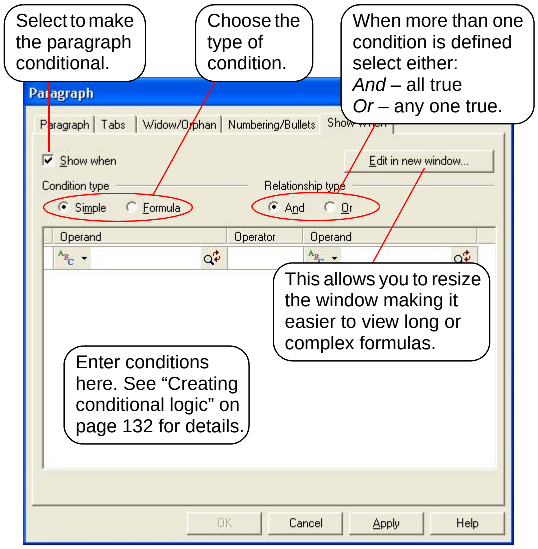

No widow or orphan control causes an orphan line.
Orphan control set to '2' ensures that at least two lines are kept together.

Conditional paragraphs

Paragraphs can be made conditional so that they appear when certain criteria are met, for instance, you may want to show a paragraph only if a customer has a certain type of account. You will typically want to use values from data fields and variables that have been made available to the publication as part of the condition.

Note that if you make a paragraph conditional it will not be shown in the editor if the specified condition evaluates to *false*. To change the settings you will need to use the logic map to work with the relevant paragraph object. See “Using the logic map” on page 92 for more information.

▶ **To make a paragraph conditional:** on the *Format* menu option click *Paragraph*. On the *Show When* tab of the *Paragraph* dialog box check the *Show When* option and enter the condition as required.



Select to make the paragraph conditional.

Choose the type of condition.

When more than one condition is defined select either:
And – all true
Or – any one true.

This allows you to resize the window making it easier to view long or complex formulas.

Enter conditions here. See “Creating conditional logic” on page 132 for details.

☑ Show when

Condition type: ● Simple ○ Formula

Relationship type: ● And ○ Or

Paragraph
Paragraph | Tabs | Widow/Orphan | Numbering/Bullets | Show When
Edit in new window...
Operand | Operator | Operand
OK
Cancel
Apply
Help

Importing paragraph content

You can include RTF content created outside DOC1 in your design layout. You can import the contents of an RTF file directly into your document, and also copy and paste the RTF – see “Using the clipboard” on page 30 for details.

The text, fonts, colors, tabs and justification used in the RTF are included when it is imported. Windows supported images, either embedded or referenced, are also included.

ⓘ RTF content can also by inserting into documents by using an RTF field. For more information, see “RTF fields” on page 108.

Text in RTF format may be slightly different depending on which application it was created by. Should you experience inconsistencies, open the RTF file in Windows WordPad and save it (as an RTF file). You can then import from this file.

▶ **To import RTF content:** from the *Insert* menu select *Presentation Objects/RTF*. Browse to the RTF file and click *Open*.

Symbols

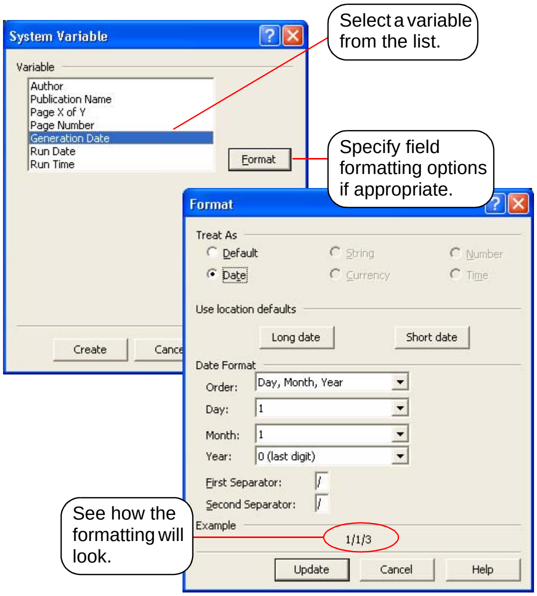
Although most commonly used characters and symbols can be entered from the keyboard, some required several key presses, for example, to type in the registered mark ® in most common fonts, you would have to type Alt+0174. A menu option is provided which accesses the Windows Character Map, where you can choose the character or symbol you want to enter.

▶ **To insert a symbol:** on the *Insert* menu click *Symbol* or click the toolbar button. In the Character Map select the required font and choose the characters you want. Copy them and then paste them into your document.

System variables

A system variable allows you to insert dynamic information that will be resolved at production run time, for example the system date or the current page number being processed. Some options generation date or page numbers. It can be formatted in various ways, depending on the information being shown, for instance, the page number can be Arabic numbers (1,2,3…), Roman numerals (i,ii,iii…), or alpha (a,b,c).

▶ **To insert a system variable:** on the *Insert* menu click *Field/System Variable* or click the toolbar button. Select the required variable from the *Variable* list and specify any options that are required. You can format certain variables by clicking the *Format* button – see “Formatting values” on page 106 for more information.



Variable
- Author
- Publication Name
- Page X of Y
- Page Number
- Generation Date
- Run Date
- Run Time

Treat As
○ Default ○ String ○ Number
● Date ○ Currency ○ Time

Use location defaults
Long date Short date

Date Format
Order: Day, Month, Year
Day: 1
Month: 1
Year: 0 (last digit)
First Separator: /
Second Separator: /
Example 1/1/3

Select a variable from the list.

Specify field formatting options if appropriate.

See how the formatting will look.

System Variables in paragraphs	
Author	The user that created the publication being processed.
Generation date	The system date when the HIP file was created for the production job by a publish task.
Page number	The current page number. Select whether you want the page number within the entire *job* or within the current *publication* or *document*.
Page count	The total number of pages in the current publication.
Page X of Y	The current page number out of the total number of pages in the publication being processed, for instance: 'Page 2 of 5'. *Page X of Y* is only displayed correctly when used in paragraphs that are left, right or center justified. The formatting of this option can be modified via Locales in the Environment tab.
Publication name	The name of the publication being processed.
Run date/time	The system date or time when DOC1 Generate began processing the production job.

ⓘ An extended range of system variables are available for use as values within other objects. See “System variables” on page 155 for details.

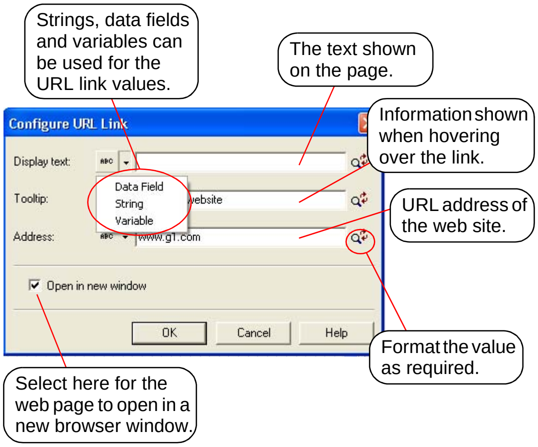
URL links

URL links can be included in applications that are intended for web enabled output. If they are used in non–web enabled output then the display text or graphic is still shown, but the link will be inactive.

Strings, variables and data fields can be used for the URL link values. For information on data fields, see “Using data fields within text” on page 73 and for details on variables, see “Variables” on page 110.

ⓘ NO CHECK IS MADE THAT THE URL LINK REFERENCES A VALID ADDRESS.

▶ **To create a URL link:** in the *Insert* menu click *Field/URL Link*
– or –
click the toolbar button.
In the *Configure URL Link* dialog box specify the options for the link.



Strings, data fields and variables can be used for the URL link values.

The text shown on the page.

Information shown when hovering over the link.

URL address of the web site.

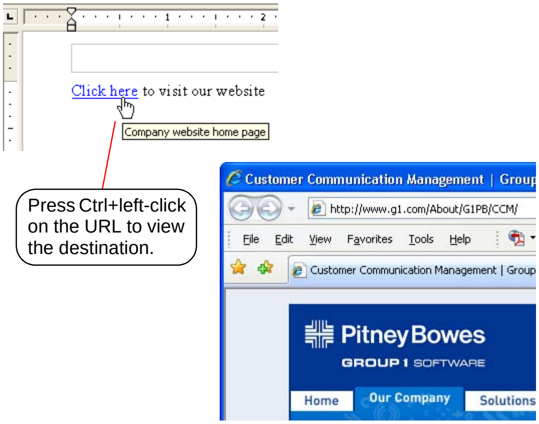
Format the value as required.

Select here for the web page to open in a new browser window.

▶ **To create a URL link for an image:** select the image and from the *Format* menu click on *Shape*. In the *Image* tab, select *Clickable image* and specify the options for the link.

▶ **To edit a URL link:** highlight or click in the URL and in the *Format* menu click *Link*. Change the options for the link as required in the *Configure URL Link* dialog box.

▶ **To test a URL link:** hover over the link and hold down Ctrl and click the left mouse button. Your internet browser will start up with the URL in the address.



Click here to visit our website
Company website home page
Customer Communication Management | Group
http://www.g1.com/About/G1PB/CCM/
File
Edit
View
Favorites
Tools
Help
Customer Communication Management | Group
Pitney Bowes
GROUP 1 SOFTWARE
Home
Our Company
Solutions

Press Ctrl+left-click on the URL to view the destination.

Working with graphics

In a document design you can reference image files created elsewhere or you can draw simple graphics using the editor's drawing tools. These are known collectively as shapes.

Image files imported from your output device environment are treated the same as locally generated images.

Text boxes are treated as graphics. The rectangle shape of a text box can be filled and given a border as for other graphics and it has the same controls for flowing into paragraph text. Unlike other graphics however, a text box is intended to contain other presentation objects.

ⓘ All local images used by a publication are automatically converted to a format suitable for your target output device at publishing time. These are stored as part of the HIP file used by DOC1 Generate.

You can control how graphics interact with paragraphs by specifying how text flows in relation to individual graphics and how much margin space is required where text flows around.

You are able to specify the order in which overlapping graphics are presented and can use the tools provided to align and space graphics in relation to one another.

Anchors and positioning

A graphic is always positioned in relation either to a paragraph or to a page; this is known as its anchor. There are three different types of anchor – inline, anchored and fixed.

When a graphic is inserted it is initially anchored according to the type of graphic. When you move the cursor over the graphic the type of anchor used is displayed. You can change the anchors of individual graphics once they have been inserted.

The following anchor types are available:

In-line Positioned as a character within paragraph text. Paragraph line spacing will be increased to accommodate the graphic, unless it has been set to an 'exact' value. In-line is the default for images.

Anchored Positioned relative within the main document flow. In the vertical plane the object is offset in relation to the previous paragraph or the top of the current container if no paragraph exists. In the horizontal plane the offset is always in relation to the left side of the container (page, text box, etc.). Anchored is the default for shapes, such as lines, rectangles, ovals and text boxes.

Fixed Positioned at a specified offset in relation to the top left corner of the page where it is inserted or multiple pages if used on the background. Fixed is the default setting, and only anchor option, for images inserted on the background.

ⓘ Note that not all anchoring methods are applicable to all types of graphic.

▶ **To change the anchor type:** select the graphic on the page
– or –
highlight the graphic in the logic map.
From the *Format* menu select *Shape*. In the *Layout* tab under *Flow* select the required anchor type and click *OK*.

Co-ordinates and dynamic positioning

Graphics using anchored or fixed methods are typically positioned by dragging the object to the required position in the page layout or by specifying co-ordinates as constant values using the shape properties, see “Moving and resizing” on page 55 for details. You can also use any object or function that supplies a value to specify co-ordinates. Where such values are provided by variable data such as data fields it follows that the graphic can have dynamic offsets that differ from document to document.

ⓘ When positioning graphics using variable data it is vital to carry out the appropriate level of testing to ensure correct placement in the production environment. You can specify how errors are handled when a shape is placed outside the page area. See “Error Handling” on page 343 for details.

▶ **To position using co-ordinates:** select the shape in the editor and on the *Format* menu click *Shape*. In the *Layout* tab select the required unit of measurement and select whether to measure from the edge of the *page* or the current *column*. Choose the appropriate value object using the value option button for the X and Y offsets. See “Working with values” on page 104 for details.

Drawing shapes

You can create simple graphics directly in the editor by using the shape drawing tools. These include lines, rectangles and ovals.

You can customize the line thickness, shading, fill color, etc. used for such shapes. When you create a new shape the attributes of the previously created shape will be used. You can adjust these by editing the shape's attributes once it has been created.

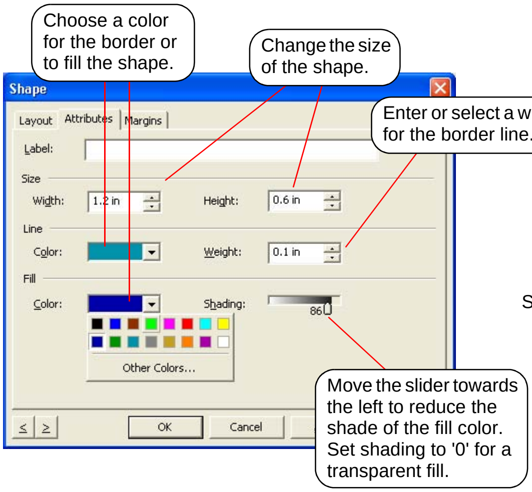
Graphics are, by default, anchored to a paragraph. However you can change this once the graphic has been drawn, see "Anchors and positioning" on page 48.

> ⓘ IMPORTANT: RESTRICTIONS INHERENT IN SOME OUTPUT DATASTREAMS MEAN IT MAY BE NECESSARY TO DOWNGRADE OR OMIT SOME TYPES OF SHAPES DEPENDING ON YOUR TARGET OUTPUT DEVICE.

▶ **To draw a shape:** click in the paragraph that you want the graphic anchored to and on the *Insert/Presentation Objects* menu select the required shape or click the appropriate toolbar icon. Click where you want to start drawing and drag out the shape to the required size. Note that holding down SHIFT while drawing will create a square, a circle, or a horizontal or vertical line.

- ╲ *Line*
- □ *Rectangle*
- ▢ *Rounded Rectangle*
- ○ *Oval*

▶ **To change shape attributes:** select the shape in the editor and on the *Format* menu click on *Shape*. Use the *Attributes* tab to set the required color, shading and line settings
– or –
use the toolbar options.



Label:

Width: 1.2 in

Height: 0.6 in

Color:

Weight: 0.1 in

Color:

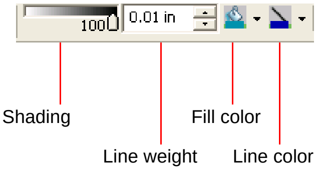
Shading: 86

Choose a color for the border or to fill the shape.

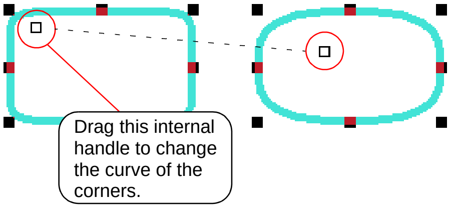
Change the size of the shape.

Enter or select a width for the border line.

Move the slider towards the left to reduce the shade of the fill color. Set shading to '0' for a transparent fill.



▶ **To change the corners of a rounded rectangle:** select the rectangle and drag the internal handle. Both the amount and the shape of the curve will change.



Drag this internal handle to change the curve of the corners.

Using images

You are able to reference images that are already stored within the Work Center repository or include new images that are available to the local file system. If you include a new image it will be automatically added to the repository but will only be available to the current document unless you specifically make it *public*. Note that in some installations these facilities may be restricted by the rights specified in your user profile.

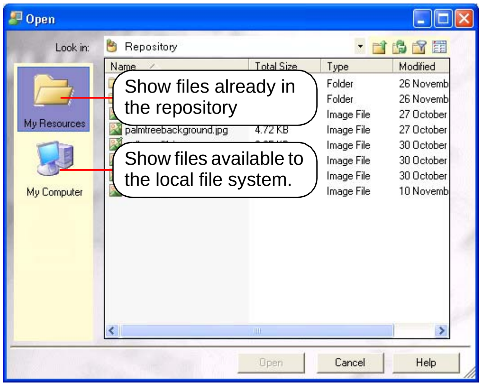
If the resolution is not already specified in an image, a default resolution of 96 dpi is used when it is inserted into a document. Images are, by default, anchored inline within a paragraph. However you can change this once the image has been inserted, see “Anchors and positioning” on page 48.

An image can also be used as a URL link (for PDF, HTML and eHTML datastreams only) by selecting the *Clickable image* option. See “URL links” on page 47 for details.

If you are working with more than one language you can change the image according to the active locale by selecting the *Multiple languages* option. See “Localizing document designs” on page 184 for details.

▶ **To insert an image:**

1. Click in the paragraph where the image is to be inserted.
2. From the *Insert/Presentation Objects* menu select *Image*
   – or –
   click the *Image* button in the Objects toolbar.
3. Use the *Open* dialog box to select the image you want to insert.



Name	Total Size	Type	Modified
		Folder	26 Novemb
		Folder	26 Novemb
		Image File	27 October
palmtreebackground.jpg	4.72 KB	Image File	27 October
		Image File	30 October
		Image File	30 October
		Image File	30 October
		Image File	10 Novemb

Show files already in the repository

Show files available to the local file system.

▶ **To change the image used:** select it and from the *Format* menu choose *Shape*. Select the *Image* tab in the *Image* dialog box and enter or browse for a different image.

Images can also be inserted using a keyed object. This allows different images to be called depending on the value of a key, for example a data field – see “Keyed objects” on page 158 for details.

Overlays

Overlays can be inserted into your document in the same way as images. However you should bear in mind that they can only be placed in a fixed position on the background layer of your document – see “Background” on page 62 for more information.

▶ **To insert an overlay:**

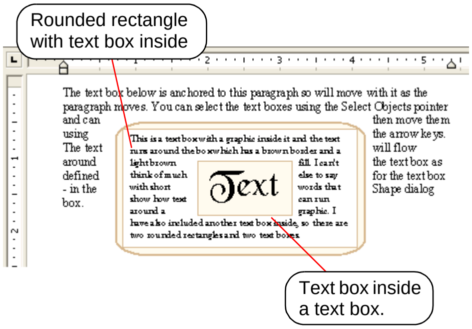
1. In the Page Setup view click in the background where the overlay is to be inserted.
2. From the *Insert/Presentation Objects* menu select *Overlay*.
3. Use the *Open* dialog box to select the overlay you want to insert.

Text boxes

A text box acts a container for specific page content and can contain both paragraph text and graphics. You can use a text box simply to highlight particular content or as a reserved area which other text can flow around.

Text boxes are rectangular shapes and are treated like graphics for the purposes of positioning and customizing. They can be inserted anywhere in a document design, for instance, in the body area, on the background, in headers and footers and even inside other text boxes.

A text box has a minimum size which is determined when it is first drawn. As you are editing, it will automatically grow downwards to accommodate content that does not fit this initial size. However, it is important to note that if your text box contains variable data, it will not be resized in the production environment – any content that overflows the available space will be lost.


Rounded rectangle with text box inside
Text
Text box inside a text box.

The text box below is anchored to this paragraph so will move with it as the paragraph moves. You can select the text boxes using the Select Objects pointer and can then move them using the arrow keys. The text will flow around the text box as defined for the text box - in the Shape dialog box.

This is a text box with a graphic inside it and the text runs around the box which has a brown border and a light brown fill. I can't think of much else to say with short words that show how text can run around a graphic. I have also included another text box inside, so there are two rounded rectangles and two text boxes.

The position of a text box (anchored to a paragraph) within the document flow may itself be affected by variable content that precedes it. For instance, if you place a text box after a transaction table, its vertical start position will be influenced by the number of transactions in the table. If there is insufficient space to fit the text box at the bottom of the current flow area – typically a page – it will, by default, be moved to the following page.

If you want the text box to appear only when there is sufficient space in the current flow area, then set the *Do not place where an overflow would occur* option. This causes the text box to be discarded when it doesn't fit.

Text boxes are, by default, anchored to a paragraph. This can be changed once the text box has been drawn; see "Anchors and positioning" on page 48.

▶ To create a text box:

1. Click in the paragraph where you want the text box anchored and from the *Insert/Presentation Objects* menu select *Text Box*
   – or –
   use the *Text Box* button in the Objects toolbar.
2. Create the box.

   If you are using a paragraph or fixed anchor click in the page layout where you want to start drawing and drag out the shape to the required size.

   If you are using an inline anchor, a default sized shape will be created within the paragraph ready for you to adjust.
3. Add content to the text box.

   For paragraph text click inside the box and start typing. Create graphics, including other text boxes, inside the box as normal.

▶ To change text box line and fill settings: select the text box in the editor and on the *Format* menu click on *Shape*. Use the *Attributes* tab to customize the settings. See "Drawing shapes" on page 50 for more information.

The look of a text box – lines, fill etc. – can be changed during production. For instance, you might want a different color for a text box that contains a negative value. This involves the use of the Action option and would generally be in some kind of conditional logic. When the condition is met, the change is applied, otherwise the original settings are used.

To change the look of a text box dynamically:

1. In the logic map mark the insertion point in the text box to be changed and from the *Insert* menu select *Control Objects/Action* .
2. In the *Action* dialog box select one of the following from the list and change the setting as required.

   Change border color
   Change fill color
   Change shade percent
   Change side thickness

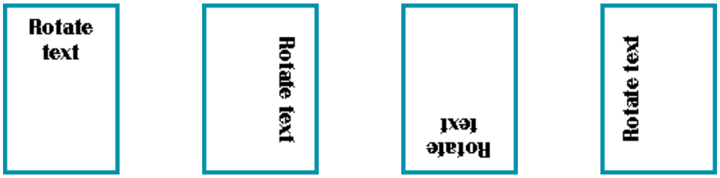
Rotating text

Text that is inside a text box can be rotated. It is rotated 90˚ clockwise, and can be rotated more than once until it is in the required plane. Text in headers and footer can also be rotated, see “Headers and footers” on page 113.

The text box itself will not be rotated, although it will adjust to cater for extra lines. However, you may need to resize it to the required dimensions once the text has been rotated. If you add more text, the text box will grow to accommodate any extra lines, until the margin is reached.

Any image or graphic that is also in the text box will not be rotated. You will have to provide it in the required rotation.

▶ **To rotate text:** place the pointer inside the text box and on the *Format* menu click on *Rotate*.– or use the *Rotate* button in the Text Formatting toolbar.



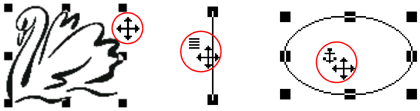
Rotated once

Rotated twice

Rotated three times

Selecting and adjusting graphics

In order to adjust a graphic, you must first select it. When it is selected, handles are displayed on the corners and sides of a bounding box. The pointer will indicate how the graphic is anchored.



Anchored fixed

Anchored in-line

Anchored to a paragraph

You can also select an existing graphic from the dialog box, cycling through them in the order they were created. This means that when you have changed the attributes for one graphic, you can move on to another one and change it without closing the dialog.

Note that white handles indicate that the graphic is open for input, e.g. a text box is ready for you to type in. Black handles indicate that the graphic is just selected, and can be moved, or if appropriate, resized.

▶ **To select a graphic:** click anywhere in the graphic or on its border.

▶ **To select multiple graphics:** select the first graphic and then hold down SHIFT whilst you click the others
– or –
use the *Select Objects* ↖ toolbar button and drag across all the graphics, ensuring that you enclose them all completely.

▶ **To view the next or previous graphic:** on the dialog box use the next > or previous < buttons.

▶ **To select all graphics:** click the *Select Objects* ↖ toolbar button to turn the select mode on and on the *Edit* menu click *Select All*.

▶ **To deselect a graphic when many are selected:** hold down SHIFT and click the graphic. The handles will disappear from the graphic when it is deselected.

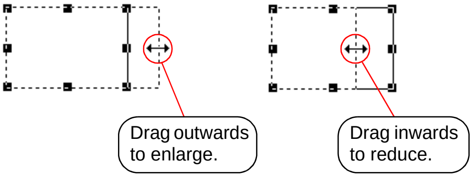
Moving and resizing

Graphics other than those using an inline anchor can be moved either directly within the editor window, or with precision using the settings in the dialog box. Graphics can be resized in the same way.

You can also align graphics to each other, or move them so that the space between them is evenly distributed.

ⓘ If you want to move graphics in relation to text, rather than other graphics, then you must use the text runaround properties. See “Layout” on page 57.

▶ **To move a graphic:** hover over the graphic and when the movement pointer ✥ displays, drag to its new position
– or –
select it in the editor and use the arrow keys to nudge it
– or –
select it in the editor and on the *Format* menu click *Shape*. Under the *Layout* tab enter the new position for the top left corner of the graphic. You can also use any object or function that supplies a value to specify co-ordinates to position the graphic, see “Co-ordinates and dynamic positioning” on page 49 for details.

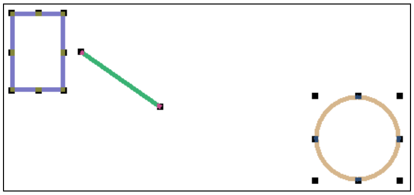


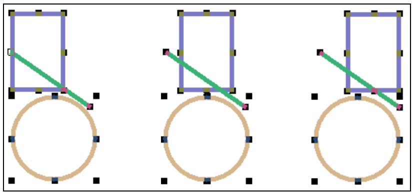
▶ **To resize a shape:** select the graphic and drag any of the handles to the new size.
– or –
on the *Format* menu click *Shape*. Under the *Attributes* tab enter the new *Width* or *Height* amounts as appropriate. If you enter a value of zero for the width or height of a line, then the line will become horizontal (width=0) or vertical (height=0).
Note that you cannot resize referenced images in the editor. If you need to do so, first resize the original using a suitable graphics program or utility and then reference the new resource in the editor.

Aligning

If you have several graphics on a page, you can line them up with each other. They can be aligned on their edges or so that the middle of each graphic lines up.

All graphics that are selected will be aligned with each other. They are aligned inside an imaginary box drawn around all the selected graphics, for instance, aligning graphics on the left will place them against the left of this box; aligning them in the center will align the centers of each graphic in the center of the box. It is important to ensure that the relevant graphic is in the correct position before alignment, for example, the right most graphic when aligning right.


Shapes selected for alignment


Aligned left Aligned center Aligned right

▶ **To align graphics:** select the graphics and on the *Format* menu click *Align or Distribute* and select one of the following alignment options:

Left – align with the leftmost graphic

Centered between the leftmost and the rightmost graphics

Right – align with the rightmost graphic

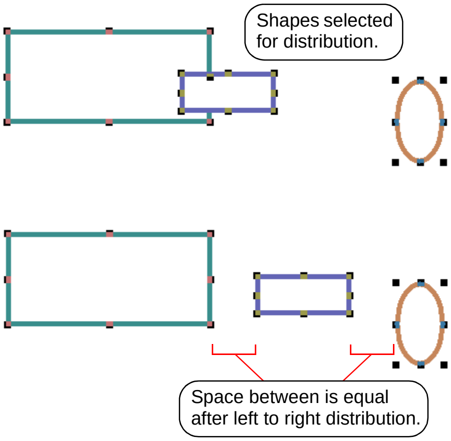
Top – align with the topmost graphic

Middle – between the topmost and the bottommost graphics

Bottom – align with the bottommost graphic

Distributing

Graphics can be distributed so that the space between each graphic is the same. The graphics on the outside stay in the same position, and all those in between are moved so that the space between the borders of each graphic is evenly distributed. This is only available when there are three or more graphics selected.



Shapes selected for distribution.

Space between is equal after left to right distribution.

▶ **To distribute graphics:** select the graphics, and on the *Format* menu click *Align or Distribute* and select one of the following distribution options:

Distribute horizontally – left to right

Distribute vertically – top to bottom

Layout

Depending on the type of anchor, you can choose how adjacent text should flow around the graphic or whether the text should appear behind or in front of it, and its alignment across the page.

Graphics that are anchored inline are treated as just another character in the text and the line spacing will normally be increased automatically to accommodate them.

▶ **To select the runaround properties for a graphic:** select the graphic and on the *Format* menu click on *Shape*. On the *Layout* tab click the required *Wrapping style*:

▣	*Square*	Text will flow evenly around the graphic
▣	*Top and Bottom*	Text will flow down to the graphic and continue after it.
▣	*Left*	Text will only be placed to the left of the graphic.
▣	*Right*	Text will only be placed to the right of the graphic
▣	*Behind*	The graphic will sit on top of any text.
▣	*In Front*	Text will flow over the graphic.

▶ **To select the alignment for a graphic:** select the graphic and on the *Format* menu click *Shape*. On the *Layout* tab click the required *Horizontal alignment*:

Other The graphic will be aligned according to the co-ordinates specified in *Position*.

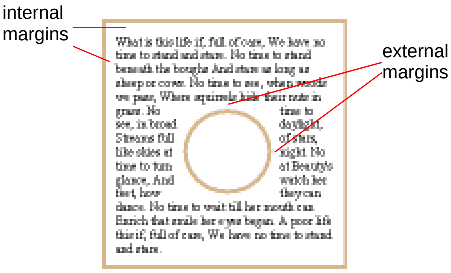
Left The graphic will be aligned with the left margin of the paragraph, with text flowing to the right.

Center The graphic will be placed centrally between paragraph margins, with text flowing to the right and to the left.

Right The graphic will be aligned with the right margin of the paragraph, with text flowing to the left.

Runaround margins

When a graphic is placed on the page you can specify how far away any text that flows around it should be. The margins and the runaround properties together determine how the graphic sits within the content of the page. Some graphics only have external margins, but shapes that can contain text – e.g. text box, address block, header, footer – have internal margins as well. These determine how far text and graphics that are inside the graphic are from the border.



▶ **To set the margins of a graphic:** select the graphic and on the *Format* menu click on *Shape*. On the *Margins* tab enter the required *Internal* or *External* margins.

Order of graphics

As graphics are inserted onto the page they are initially layered in the order they are placed on the page. This means that later graphics will be displayed over earlier graphics if they overlap. You can adjust this order as required.

You cannot move graphics in front of or behind text in this way. You must use the runaround properties of the graphic to do this – see “Layout” on page 57

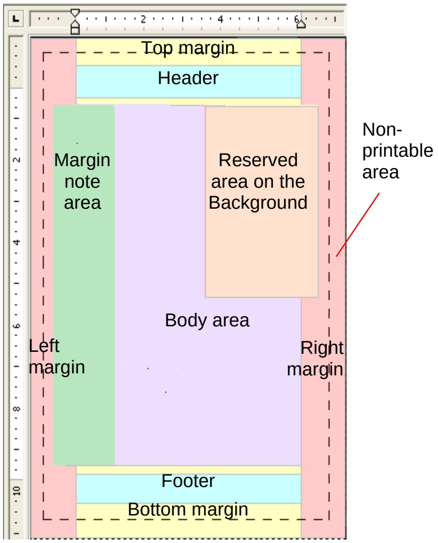
Note that you can cycle through graphics in their 'layer' order by using the previous < and next > buttons at the bottom of the *Image* dialog box.

▶ **To move a graphic forwards or backwards one layer:** select the graphic and on the *Format* menu point to *Order* and select either *Bring Forward* or *Send Backward* .

▶ **To move a graphic in front of or behind all other graphics:** select the graphic and on the *Format* menu point to *Order* and select *Bring To Front* or *Send To Back* .

Page layout

The primary area where content is added to document designs is known as the **body** which consists of one or more **columns** within the body area. Content will automatically flow onto new columns and pages as required. The space available for the body is constrained by the overall page size you specify and the position of any other layout areas such as the **margins, headers and footers** that you define.



Documents also have a **background** layer, which you can use to:

- add content 'behind' the body for example, to provide a watermark or similar
- include a **reserved area**, to specify that part of the page area is unavailable for content in the body area. It is possible to update the content of a reserved area dynamically if required.
- include a **margin note area** that is used to control the placement of margin notes. For more information see “Working with margin notes” on page 119.

Headers and footers can be used to specify content that is to be repeated above and below the body area. They can also be used above and below individual columns.

In all these areas of the page you can create **text boxes** that act as a container for specific content. You can use a text box simply to highlight particular content or as a reserved area around which text will flow.

Page setup

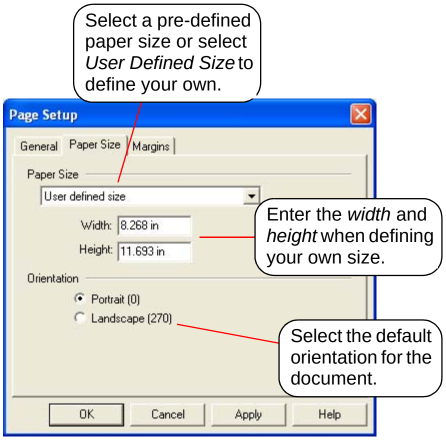
The *Page setup* view – available from the *View* menu – allows you to specify the various options that define the look and layout of the page, such as margins, background, headers and footers.

Page size and orientation

The page size is set for a complete document in the Page Setup view. This means that initially all pages in a publication are the same size. If your publication requires more than one page size, then you must create a new document for each different page size.

Unlike page size, you can have different page orientations within a single document. To change the orientation of a page you must use a document section. For more details, see “Sections and page control” on page 122.

▶ **To set the page size:** select *View/Page Setup* and double-click Page Setup in the logic map. The *Page Setup* dialog box is displayed. Click the *Paper Size* tab and apply the necessary changes.



Paper Size: User defined size

Width: 8.268 in

Height: 11.693 in

Orientation

● Portrait (0)

○ Landscape (270)

Select a pre-defined paper size or select *User Defined Size* to define your own.

Enter the *width* and *height* when defining your own size.

Select the default orientation for the document.

Margins and body area

Margins are the space around the edge of the page that are not available as part of the body area. They are measured from the edge of the page and you should set the size of the margins to meet the requirements of your document designs.

If your design incorporates a booklet format, you can use the *Alternate left and right margins* option to ensure that the pages’ margins mirror one another when printed. The settings are based on the first (left sided) page in your publication.

In addition, many printers have an area around one or more edges that cannot be printed on. You can specify dimensions for this along with your user defined margins so that your page design keeps within the printable area.

By default, the area available for the main document body is automatically defined by the space available once margins, page headers and footers, and any reserved areas have been taken into consideration. If you want to change the body area dimensions, change any of these.

▶ **To define margins:** select *View/Page Setup* and double-click Page Setup in the logic map. The *Page Setup* dialog box is displayed. Click the *Margins* tab and apply the necessary changes.

Background

For each page in the document view there will be a corresponding page in the background layer. These pages display the content from the document foreground grayed out beneath the background objects for that particular page.

By default, content on the background layer appears on all pages of the document although you can use conditions to restrict objects to particular pages if required. For instance, if you place an object on the third background page it will appear on every page of the document unless you create a condition to limit it to a specific page.

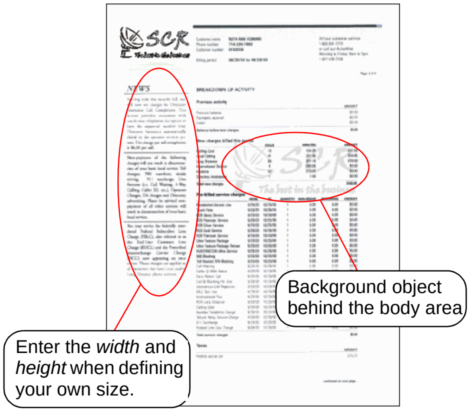
All objects on the background layer have a fixed position. You can place graphics directly on the background. All other presentation objects will need to be included in a text box that establishes the required position on the page. Objects on the background layer can use the entire page area if required and are not restricted by margins.

Where background content appears in the same place as content in the body area, the background material will be placed 'behind' the other content. You may want to use this to achieve a watermark effect or similar.

A **reserved area** is a special object only available to the background. It defines space on the page that cannot be used by content within the main document body and this will flow around the reserved area as necessary.

You can use any normal presentation feature directly within reserved areas. Additionally, you can dynamically add or replace the content within a reserved area within the main document logic.

For information about creating and updating reserved areas see “Working with reserved areas” on page 117



Background object behind the body area

Enter the *width* and *height* when defining your own size.

▶ **To work with the background:** click the toolbar button
– or –
from the *View* menu choose *Page Setup*.

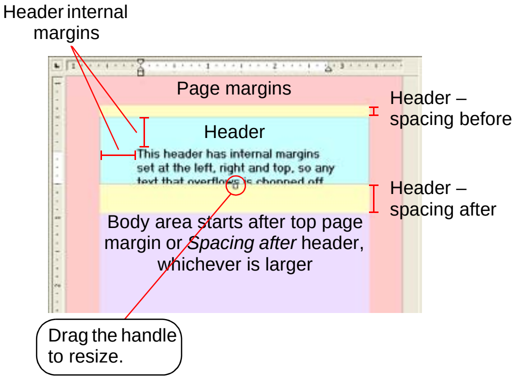
▶ **To add text to the background:** create a text box, position it as required and then type text into it. See “Creating paragraphs” on page 33.

▶ **To add graphics to the background:** use the available background area to add graphics as required. See “Working with graphics” on page 48 for details.

A **margin note area** is a defined space in the background, which is used to control the positioning of margin notes. See “Working with margin notes” on page 119 for more details. Multiple margin note areas can be defined in the *Page Setup* view.

Page headers and footers

Headers and footers contain content to be placed above and below the body area on each page of a document. You may, for instance, want to add a paragraph that includes the date in the page header or use a footer that shows the current page number.


Header internal margins
Page margins
Header – spacing before
Header
Header – spacing after
Body area starts after top page margin or *Spacing after* header, whichever is larger
Drag the handle to resize.

This header has internal margins set at the left, right and top, so any text that overflows is chopped off

Headers and footers are placed before and after the body area and are constrained by the left and right margins. You must ensure that they are large enough to cope with the expected header and footer content. They are defined as part of the background layer.

The start and end of the body area are defined by the top/bottom page margins or the header/footer, whichever is larger.

Content defined for headers and footers does not have to appear on all pages. You can place a paragraph conditionally depending on the current page number or other criteria. You can also use conditional objects with header/footer logic. See “Page and job handling” on page 113 for details.

▶ **To work with page headers and footers:** change to the *Page Setup* view and on the *View* menu choose *Header* or *Footer* as appropriate. Enter text and graphics into the header or footer as normal.

Note that the combined size of the header and footer cannot be more than 75% of the page, thus leaving a minimum of 25% for the body area.

▶ **To resize a page header or footer:** click in the area and drag the appropriate handle as required
– or –
in the Page *Setup* logic map, highlight the header or footer and from the *Format* menu select *Shape*. On the *Header and Footer* page, set the *Custom height* as required.

You can adjust header or footer attributes as with any shape object. See “Working with graphics” on page 48 for details.

Columns

By default DOC1 uses a single body area for positioning main document content. If required you can split this into multiple columns. Such layouts are often known as newspaper columns. Content that is designed to appear within the main document body will automatically flow through the available column areas, top-to-bottom and left-to-right or, if required, right-to-left.

TREE TOP INVESTMENTS

TREE TOP INVESTMENTS

You can use the same layout for columns on all pages, use different layouts for different pages or use a combination of column layouts on a single page.
For instance, the first page might have a paragraph of introduction at the top that spans the page; two or more columns may start part way down; the middle pages might contain just columns and the last page may have information following the columns such as a payment slip.

When working with multiple columns content is created in the normal way and there are no restrictions on the type of content you can include. Paragraph text will wrap within the available column width as with a single body area and will flow around anchored graphics and background items as normal.

ⓘ Care should be taken when using anchored graphics to ensure that they fit as required with the column boundaries.

▶ **To use columns:** click in the paragraph where you want the columns to start and use the *Columns* toolbar button, choosing the number of columns required. This will create a basic column layout from the current position with no headers or footers, and equal width columns on the whole available page.

ⓘ You can use the logic map to specify more advanced column layout requirements – see “Customizing columns” on page 124 for more information.

Using tables

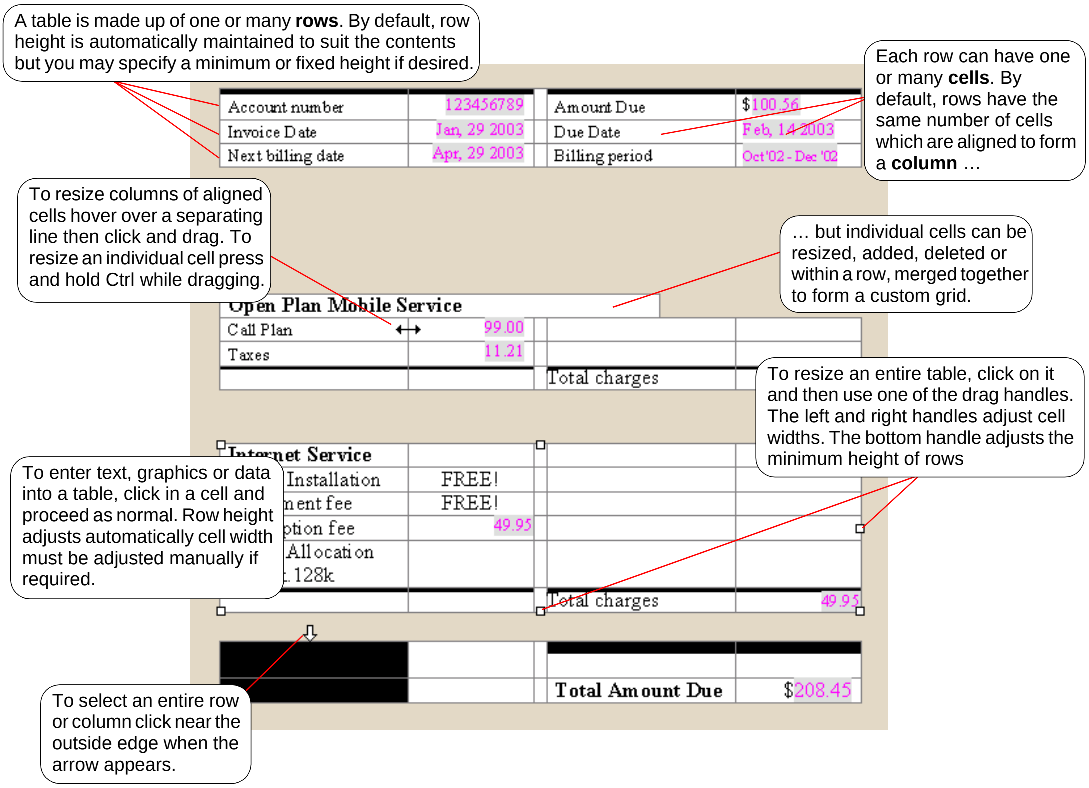
DOC1 uses tables for two distinct functions:

A static table
is used to align text and graphics in rows and cells or generally to help with positioning objects on a page (as with most word processing tools).

A transaction table
is used to control how repeating data is presented, i.e. how multiple iterations of the same repeating data structure are ordered and how they are positioned in relation to one another.

The layout of both types of table are defined in the same way but there are several additional settings that relate only to transaction tables.

> THIS SECTION COVERS HOW TABLES ARE CREATED AND CUSTOMIZED. SEE ALSO “REPEATING DATA AND TRANSACTION TABLES” ON PAGE 74.



Account number	123456789	Amount Due	$100.56
Invoice Date	Jan, 29 2003	Due Date	Feb, 14 2003
Next billing date	Apr, 29 2003	Billing period	Oct'02 - Dec '02

Open Plan Mobile Service			
Call Plan	99.00		
Taxes	11.21		
		Total charges	

Internet Service			
Installation	FREE!		
ment fee	FREE!		
ption fee	49.95		
Allocation . 128k			
		Total charges	49.95

		Total Amount Due	$208.45

A table is made up of one or many **rows**. By default, row height is automatically maintained to suit the contents but you may specify a minimum or fixed height if desired.

Each row can have one or many **cells**. By default, rows have the same number of cells which are aligned to form a **column** …

To resize columns of aligned cells hover over a separating line then click and drag. To resize an individual cell press and hold Ctrl while dragging.

… but individual cells can be resized, added, deleted or within a row, merged together to form a custom grid.

To resize an entire table, click on it and then use one of the drag handles. The left and right handles adjust cell widths. The bottom handle adjusts the minimum height of rows

To enter text, graphics or data into a table, click in a cell and proceed as normal. Row height adjusts automatically cell width must be adjusted manually if required.

To select an entire row or column click near the outside edge when the arrow appears.

The Table Wizard allows you quickly and easily to create the basic structure and controls for any type of table. Once created you can modify the table layout and settings as required.

▶ **To create a table:**

1. Position the cursor at the paragraph or other presentation object the table is to follow.
2. On the *Table* menu click *Table Wizard*.
3. Use the pages of the table wizard to specify the basic layout for your table.
4. Once the table has been created you can edit the row and column structure as required and specify border and shading settings for individual cells.
5. Add content to the table cells as required.

ⓘ FOR ADVANCED REQUIREMENTS SUCH AS INCLUDING A TABLE CONDITIONALLY OR AS PART OF ANOTHER OBJECT YOU MAY NEED TO USE THE LOGIC MAP WHEN CREATING IT. SEE “USING THE LOGIC MAP” ON PAGE 92 AND THE RELEVANT TOPIC IN THE “ADVANCED PUBLICATION DESIGN” SECTION.

Table positioning

A table within the body area is always positioned in relation to the object selected when it was inserted. You cannot move a table except by cutting and pasting it so that it is relative to a different object.

If you want to position a table at a fixed position on a page or at fixed offset from an object you will need to include it in a text box. Note that tables within a text box cannot grow beyond the page on which they are started.

ⓘ SEE “TEXT BOXES” ON PAGE 52 FOR INFORMATION ABOUT USING TEXT BOXES.

As with other presentation objects you can edit the table’s properties to change the amount of space to be left between the table and the objects that appear before and after it in the document flow. You can also specify that the table is to be kept together on a single page where possible or kept with the presentation object that follows it.

By default, a table is positioned as close as possible to the left hand page margin. You can specify an indent to shift it to the right if required.

▶ **To customize table positioning properties:** from the menu select *Table/ Properties*. Use the *General* page to select special positioning options.

ⓘ WIDOW AND ORPHAN CONTROLS ARE NORMALLY ONLY USEFUL WHEN DESIGNING A TRANSACTION TABLE. IF YOU DO SPECIFY THEM FOR A REGULAR TABLE HOWEVER, THE SETTINGS RELATE TO THE NUMBER OF INDIVIDUAL ROWS THAT ARE TO BE KEPT TOGETHER WHEN THE TABLE OVERFLOWS ONTO A NEW PAGE.

Basic editing

Content is added to a table by clicking in a cell then typing text, adding a data field or inserting an object as you would elsewhere.

You cannot insert the following objects within a table cell:

- Another table
- A new page action

Each table row starts off with a default minimum height. As content is added to the table, the height of individual rows will automatically grow to fit by default. You can specify that rows have a different minimum height or that they may not exceed a maximum height.

▶ **To adjust row height:** select any part of the table and from the menu select *Table/Properties/Table*. Use the *Advanced* page to select the appropriate *Row Height* setting and specify a measurement if required.

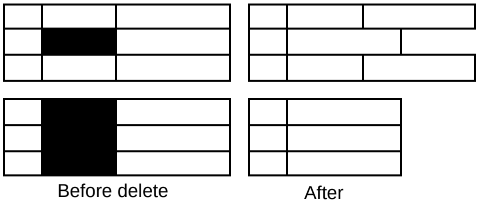
When new rows or cells are added they adopt the attributes and dimensions of the currently selected element by default.

▶ **To add a row, column or cell:** place the insertion point in an existing cell of the table, and from the *Table* menu click *Insert* and select the required row, column or cell option.

Most of the common resizing and editing operations can be performed using mouse operations. However, you may want to specify precise settings for some rows and cells.

When cells are deleted or resized, all remaining cells to their right retain their existing dimensions but close up to the left.

▶ **To resize cell widths:** select the cells to be adjusted and from the Table menu select *Properties/Cell*. In the *Cell* dialog box enter a new *Width* and click *OK*.



Adjacent cells in a row can be merged together to form a single cell (existing content will also be merged). This can be used, for example, for a description that applies to more than one column.

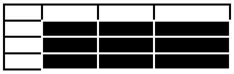
▶ **To merge cells:** select the relevant cells and from the *Table* menu click *Merge cells*.

▶ **To delete tables rows, columns or cells:** click in or select what you want to delete, and from the *Table* menu click *Remove* and select the required table, row, column or cell option.

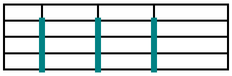
Borders and shading

You can customize the borders and shading used for cells, rows and tables. For borders you can specify the width and color of the line to be used for each edge of a cell, row or table. You can also specify rounded corners for the table. For shading, select the color to be used to fill the cell behind any contents it might have. As with all color issues, bear in mind the capabilities of your target output environment when setting these options.

The settings you specify for a selection of cells are applied individually to each selected cell.



Specifying a colored left hand border for these cells …



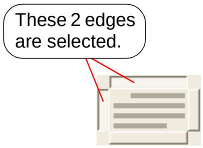
… will adjust borders for these edges.

▶ To adjust borders and shading:

1. Select the table, row or cells to be amended.
2. From the *Table* menu, select *Properties* and *Table*, *Row*, or *Cell* as appropriate.
3. On the *Style* page of the *Table Properties* dialog box specify the required color and line settings.

   For borders use the edge palette to select the edges you want to be changed. Note that any unselected edges will have existing settings removed.

   For shading, select a shading percentage for the color – 0 will be no color; 100 will be full color.



4. Click *Apply* to see the effect of your changes or click *OK* to apply the new settings and close the dialog box.

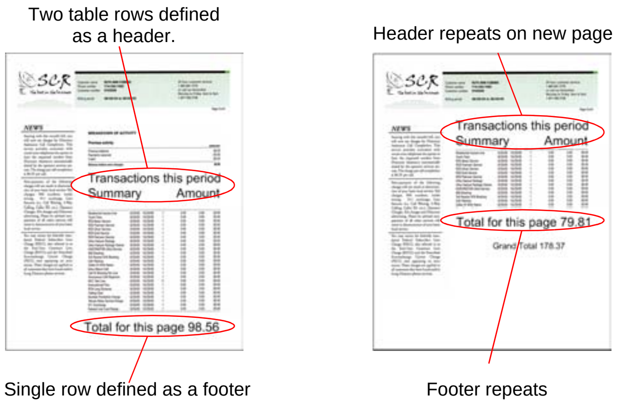
Tables can have rounded corners. They are applied to each corner of the table, on every page if the table overflows. The rounded corner is drawn in corner cells only; if the size specified is larger than the cell can accommodate, then that particular corner will not be rounded. Note that you may have to adjust the cell margin or format to prevent the contents of a cell overprinting the rounded corner.

> If you are using rounded corners on a shaded table and the output is to AFP, the shading may not be consistent with the same shading elsewhere in your publication. To ensure consistency specify CMYK or RGB color support for the AFP device – see Color support in “ Output device options for AFP” on page 282.

▶ To specify rounded corners for a table: select the table and from the *Table* menu, select *Properties*. On the *Style* page of the *Table Properties* dialog box select *Enable rounded corners* and specify the required radius of the curve.

Headers and footers

Table headers and footers should not be confused with the headers and footer objects that can be applied in Page Setup options. Using a table header means that the contents of the first row (and, optionally, other consecutive rows) will be repeated at the top of the table whenever the table contents flow to a new page. Similarly, the contents of the table footer row will be repeated at the bottom of the table on all pages in which it appears.


Two table rows defined as a header.
Header repeats on new page
Transactions this period
Summary
Amount
Total for this page 98.56
Transactions this period
Summary
Amount
Total for this page 79.81
Grand Total 178.37
Single row defined as a footer
Footer repeats

When you specify the use of headers or footers it is assumed that only the first or last row is to be used for this purpose. You can add further rows to an existing header/footer if they are adjacent to these.

▶ **To enable or disable table headers or footers:** select the Table Header or Table Footer object in the logic map and from the *Table* menu select *Insert/Row below* or *Row above*.

▶ **To add a row to a header or footer:** click on a header or footer row and from the *Table* menu select *Insert/Row below* or *Row above*. By default the new row adopts all the properties of the current header /footer row.

▶ **To remove a specific row from headers or footers:** select the row(s) and from the *Table* menu select *Remove/Row*.

Using overflow variables

The Work Center provides two system variables specifically for use within table headers and footers. *Table page overflow* and *Table column overflow* return a *true* value when the content of the table flows onto another page or presentation column, otherwise a *false* value is returned.

For more information on variables, see “System variables” on page 155.

You can use the variables with conditional logic to set the content of your headers and footers based on the value they return. For instance, in the example below when the table page overflow variable is *True* the footer displays 'Sub-total' and when it is *False* it displays 'Monthly total'.

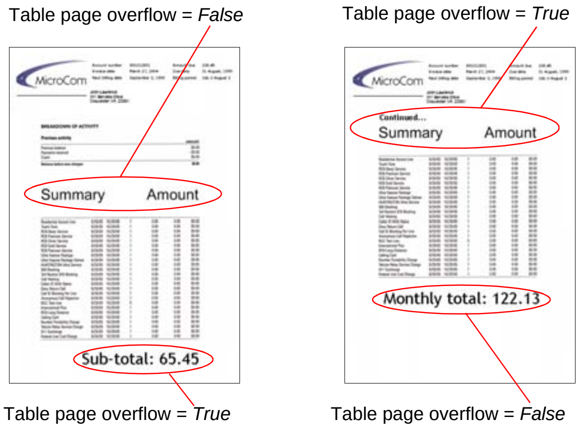


Table page overflow = *False*

Table page overflow = *True*

Table page overflow = *True*

Table page overflow = *False*

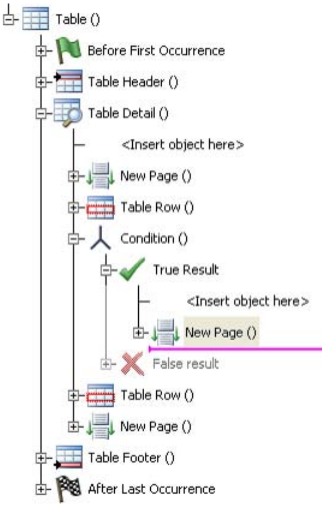
Pagination

By default tables that span pages will automatically break where the last complete row can be fitted on the page. You can change this behavior using the *Keep on same page*, *Widow and orphan processing*, *Keep transaction rows together* table properties or, alternatively, you can manually insert page throws in your table logic where required.

Controlling pagination using table logic

Use the *New Page* object to specify where a page break is to appear in your table logic. New pages can be placed anywhere within a table detail group, except inside a table row. Note that the table properties described above are overridden when this type of pagination is used.

The *New Page* dialog box is invoked allowing identification and page header/footer properties to be entered. Refer to "Generating new pages" on page 113 for detailed information.



```
Table ()
    Before First Occurrence
    Table Header ()
    Table Detail ()
        <Insert object here>
        New Page ()
        Table Row ()
        Condition ()
            True Result
                <Insert object here>
                New Page ()
            False result
        Table Row ()
        New Page ()
    Table Footer ()
    After Last Occurrence
```

Working with input data

Input data is variable information from your business environment that can be used as part of your documents.

DOC1 Generate uses an input data file that contains a set of such information for each publication to be produced. Typically this is a data base extract that has been prepared for use with DOC1.

Within design files, references to input data are created as **data field** or **data record** objects that are aliases for the actual data elements that will be available in the input data file. Such objects provide indirect references to the data so you do not need to know the exact data elements required when creating your design. The fields are mapped to actual elements of input data before the publication is put into production.

Such aliases to records and fields are themselves grouped together into **data dictionaries** that, for example, represent all the data available to a particular type of job or a sub-set of elements that are referenced frequently.

ⓘ YOU MAY HAVE ACCESS TO ONE OR MANY DICTIONARIES DEPENDING ON YOUR ENVIRONMENT AND USER PRIVILEGES. IN LARGER INSTALLATIONS DICTIONARIES WILL OFTEN BE CREATED AND MAINTAINED BY THE DOC1 ADMINISTRATOR. SEE "WORKING WITH DATA DICTIONARIES" ON PAGE 267 FOR MORE INFORMATION.

Broadly speaking there are two types of data elements: non-repeated and repeated.

Non-repeating data typically contains information such as account numbers, names, addresses and so on. Fields that contain non-repeating data appear only once in the input data for a single publication and can be used directly within paragraph text.

Repeating data typically contains transaction type information such as account entries or itemized telephone calls. Repeating data is iterated within the data for a single publication – i.e. the same data structure appears multiple times consecutively. The data structure itself can consist of one or many elements that each need to be referenced as a separate data field when used in a document design. You will normally want to use a **transaction table** to control how the iterations of such elements are placed within document designs.

Using data fields within text

Fields from non-repeating data elements can be inserted directly into paragraph text as required.

As you enter your paragraph text you can reference existing data fields or create new ones assuming you have the relevant user privileges. If you intend to create new data fields you need to be confident that appropriate data elements will be available in the input data files that will eventually be used with the DOC1 job.

References to fields can also be used to provide parameters for functions and other objects requiring values. The values provided by a field can also be stored and manipulated within a variable before being output. See “Working with values” on page 104 for more information.

When creating a data field you will need to define its expected data type (string, date, number and so on) so that the correct formatting can be applied to it. You will also normally want to supply an example of the data expected in the field so that the document view presents realistic content.

▶ To reference an existing data field:

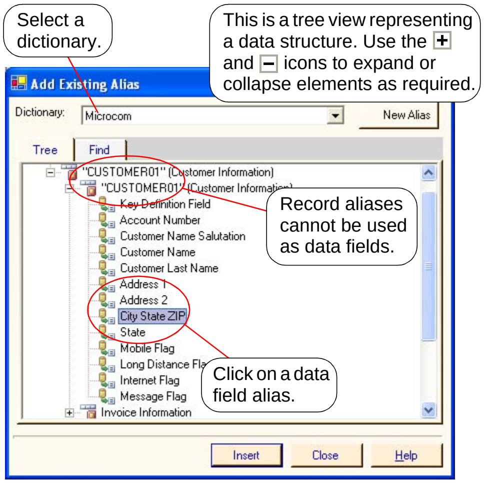
1. Position the cursor as required within a paragraph.

2. On the *Insert* menu select *Field/Data Field*.

3. In the *Add Existing Alias* dialog box select the data dictionary the field belongs to (if you have access to multiple dictionaries).

4. From the *Tree* view click on the data field that you want to reference and click *Insert*.
   – or –
   click on the data field and place it in the document using a drag-and-drop operation.

You can use *Find* to search for an existing data field. For more details, see “Working with data dictionaries” on page 267.



Select a dictionary.

This is a tree view representing a data structure. Use the ⊞ and ⊟ icons to expand or collapse elements as required.

Record aliases cannot be used as data fields.

Click on a data field alias.

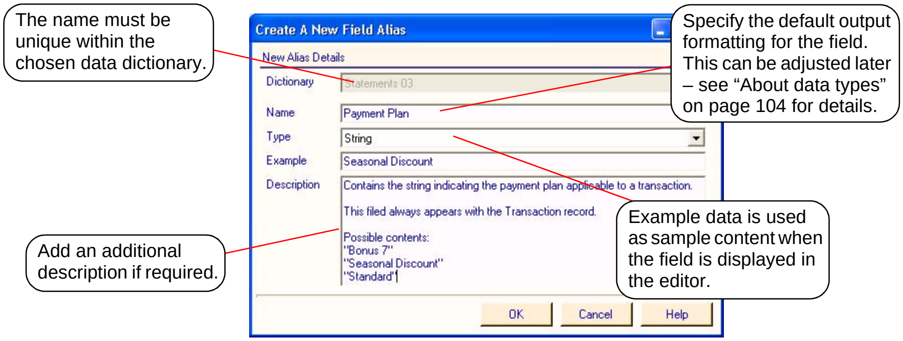
▶ To create a new data field:

1. Position the cursor as required within a paragraph.

2. On the *Insert* menu select *Field/Data Field*.

3. In the *Add Existing Alias* dialog box select the data dictionary in which the new field will be grouped. See “Working with data dictionaries” on page 267 for more information.

4. In the *Add Existing Alias* dialog box click the *New Alias* button.
5. Enter the details in the *Create A New Field Alias* dialog box and click *OK* to create the new reference.



Create A New Field Alias

New Alias Details

Dictionary: Statements 03

Name: Payment Plan

Type: String

Example: Seasonal Discount

Description: Contains the string indicating the payment plan applicable to a transaction.

This filed always appears with the Transaction record.

Possible contents:
"Bonus 7"
"Seasonal Discount"
"Standard"

The name must be unique within the chosen data dictionary.

Specify the default output formatting for the field. This can be adjusted later – see “About data types” on page 104 for details.

Example data is used as sample content when the field is displayed in the editor.

Add an additional description if required.

▶ **To change to a different data field reference:** right-click on it in the document view and select *Format* from the shortcut menu. Click the *Value* button and the *Add Existing Alias* dialog box is displayed, enabling you to change data fields.

▶ **To format data field content:** right-click on the field in the document view and select *Format* from the shortcut menu and in the *Field* dialog box, click the format button. For more detailed information see “Formatting values” on page 106. This relates to the content of the field and not how it looks on the page. You can, of course, change the font and text properties assigned to the field output in the usual way – see “Entering and formatting text” on page 33 for details.

Repeating data and transaction tables

A transaction table provides the special controls required to place data elements that appear iteratively within input data.

Like a regular table, a transaction table is created using the table wizard and its cosmetic attributes can be customized in the same way. See “Using tables” on page 65 for details.

> ⓘ A REPEATING DATA OBJECT CAN ALSO BE USED TO HANDLE ITERATIVE DATA WHERE IT NEEDS TO BE PROCESSED WITHOUT ACTUALLY PLACING FIELDS WITHIN A DOCUMENT DESIGN OR WHERE YOU NEED MORE PROGRAMMATIC CONTROL OF FIELD PLACEMENT. SEE “ADVANCED DATA HANDLING” ON PAGE 177 FOR DETAILS

The main areas where a transaction table differs are:

- you will need to specify *Control Object* i.e. identify the repeating data element to be placed within the table, or specify a repeat count.

- the number of rows specified for the table are automatically multiplied for each iteration of the repeating data that appears
- the table layout can appear within multiple presentation columns on the same page.

> IF YOU ADD DATA FIELDS THAT ARE NOT PART OF THE *CONTROL DATA* (I.E. THE REPEATING DATA) THEY WILL BE TREATED AS STATIC ELEMENTS.

Initially, only data fields from the repeating data element are added to the cells of the table as part of the wizard. Once the wizard has created the table however, you can add any other text or graphics to the table cells in the normal way. Static elements of this kind will be repeated every time an iteration of the relevant row is generated by the repeating data.

As the repeating data creates additional table rows new pages are automatically generated when the height of the table exceeds the available page space.

By default, the repeating data will be presented according to the order in which it is provided in the input data file. If required, you can specify a sort that re-sequences the data prior to being placed in the table – this is available only when a data element has been selected to control the repeat.

▶ To create a transaction table:

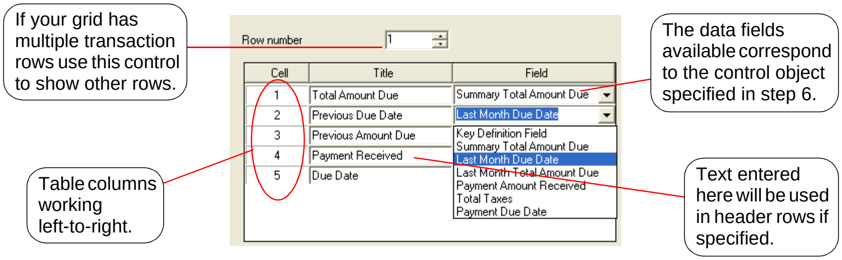
1. Make sure the cursor is positioned within the paragraph or object immediately before where the table is to be inserted.
2. On the *Table* menu select *Table Wizard*.
3. In the wizard choose *Transaction* as the table type.
4. Define a grid for the data.

   On the Rows and Columns page specify the *Number of rows* and *Number of columns* that form the required layout per transaction. Typically you will want to create a grid that has a cell for <u>one</u> iteration of each field from your repeating data.
5. Select the *header* and *footer* options if required.

   If selected an extra table row will automatically be inserted before/after the main grid. You can add additional rows to the header/footer once the table has been created.
6. Specify a control object.

   Use the Control Data page to identify the repeating data element that will be presented in the table. Using the *Select* button will invoke the *Add Existing Alias* dialog box where you can either create a new alias or select from the list of existing repeating data elements already defined in the available data dictionaries. See page 72 for more information about data aliases and data dictionaries.
7. Map fields to table cells.

   Use the Cell Contents page to identify where fields from the repeating data will be placed in the table.



If your grid has multiple transaction rows use this control to show other rows.

The data fields available correspond to the control object specified in step 6.

Table columns working left-to-right.

Text entered here will be used in header rows if specified.

Row number: 1

Cell	Title	Field
1	Total Amount Due	Summary Total Amount Due
2	Previous Due Date	Last Month Due Date
3	Previous Amount Due	
4	Payment Received	
5	Due Date	

8. Specify data sorting options if required.

   See “Sorting” on page 79.

9. Specify the layout of presentation columns.

   See the following section “Presentation”.

10. Specify general table attributes on the General Options page as required.

    Selecting *Keep on same page* will attempt to keep an entire transaction table in the same presentation column.

11. Add other data as required.

    Once the table has been created it will be populated with the specified fields from the repeating data element. Header and footer rows may also contain data. However, you will often want to add other text and graphics or fields outside the repeating data element itself. Click in a cell and then add new data as required.

12. Include any table functions as required.

    Running totals and transaction count can be included – see “Counts and totals” on page 79 for details.

▶ **To change the control data or specify advanced table controls:** right-click within the table and select *Table Properties* from the shortcut menu.

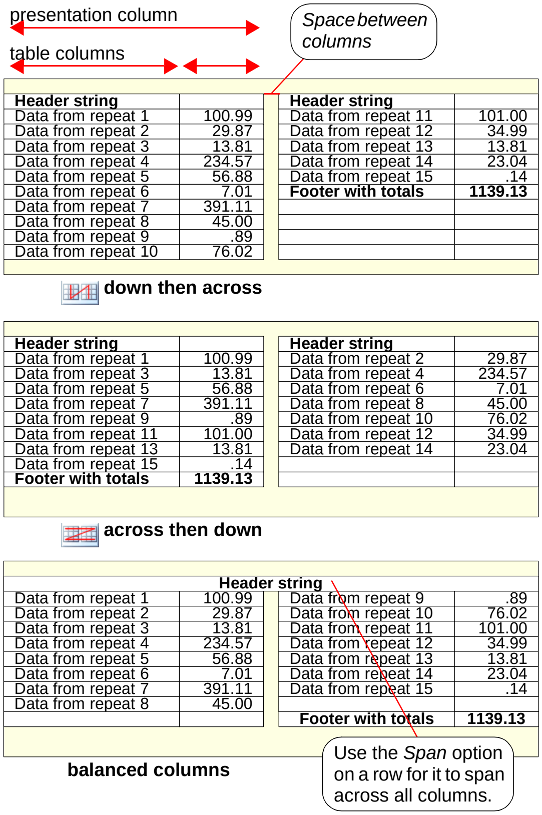
When viewing a transaction table in the editor, you can restrict the number of transactions to be shown in the document view. This could be useful if, for instance, you want to see how a table looks, without the need for real data. Use any value in the table, and set the repeat count. IMPORTANT: this has influence only during editing and has no effect on how data is handled in preview or production.

▶ **To restrict the number of transactions displayed while editing:** right-click within the table and select *Table Properties* from the shortcut menu. In the *General* page specify the *Number of transactions* as required.

Presentation

By default, a transaction table will be created within a single presentation column that fills the available page width. The table columns themselves will be of an equal size spread across the presentation area.

Table flow examples using two presentation columns



Header string		Header string	
Data from repeat 1	100.99	Data from repeat 11	101.00
Data from repeat 2	29.87	Data from repeat 12	34.99
Data from repeat 3	13.81	Data from repeat 13	13.81
Data from repeat 4	234.57	Data from repeat 14	23.04
Data from repeat 5	56.88	Data from repeat 15	.14
Data from repeat 6	7.01	**Footer with totals**	**1139.13**
Data from repeat 7	391.11		
Data from repeat 8	45.00		
Data from repeat 9	.89		
Data from repeat 10	76.02		

down then across

Header string		Header string	
Data from repeat 1	100.99	Data from repeat 2	29.87
Data from repeat 3	13.81	Data from repeat 4	234.57
Data from repeat 5	56.88	Data from repeat 6	7.01
Data from repeat 7	391.11	Data from repeat 8	45.00
Data from repeat 9	.89	Data from repeat 10	76.02
Data from repeat 11	101.00	Data from repeat 12	34.99
Data from repeat 13	13.81	Data from repeat 14	23.04
Data from repeat 15	.14		
Footer with totals	**1139.13**		

across then down

Header string			
Data from repeat 1	100.99	Data from repeat 9	.89
Data from repeat 2	29.87	Data from repeat 10	76.02
Data from repeat 3	13.81	Data from repeat 11	101.00
Data from repeat 4	234.57	Data from repeat 12	34.99
Data from repeat 5	56.88	Data from repeat 13	13.81
Data from repeat 6	7.01	Data from repeat 14	23.04
Data from repeat 7	391.11	Data from repeat 15	.14
Data from repeat 8	45.00		
		Footer with totals	**1139.13**

balanced columns

Use the *Span* option on a row for it to span across all columns.

Once the table has been created you can adjust the width of any individual table column as required and you can customize the line and fill options of all cells.

You can also specify that additional presentation columns are to be used and indicate how the repeating data and associated static elements should flow through the multi-column layout. When working with multiple presentation columns you will need to ensure that the width of your table allows enough room for the number of columns you select.

As with regular tables, a transaction table can include header and footer rows. The content of header rows is repeated at the top of each presentation column. Footer rows are presented below the last row generated by the repeating data. When multiple presentation columns are being used you can request that the header or footer rows are expanded to span across the entire set of columns if required.

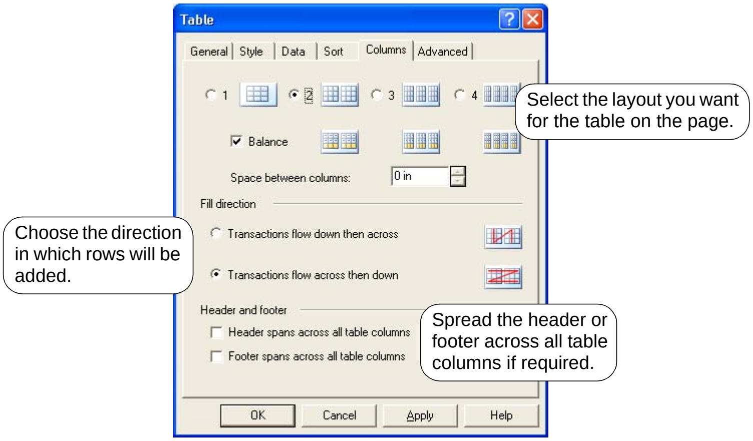
If you have entered *Title* text for any table columns this will appear in the relevant header cell by default. You can move this if required.

Widow and orphan settings for transaction tables relate to complete sets of transaction rows. For example, if the table has 3 transaction rows and you specified a widow value of 2, at least 6 rows would actually be kept together prior to a page throw.

▶ **To change or add to table data:** edit cell contents created by the wizard in the normal way. You can add, replace or delete data fields as required. You can also add any static data such as paragraph text and graphics.

▶ **To customize table size and style:** you can adjust column widths and row height settings, add or remove cells and set their borders and shading as for a regular table. See “Using tables” on page 65 for details.

▶ **To change the presentation column layout:** right-click within the table and select *Table Properties* from the shortcut menu. Use the *Columns* page as required.



○ 1 ● 2 ○ 3 ○ 4

☑ Balance

Space between columns: 0 in

Fill direction

○ Transactions flow down then across

● Transactions flow across then down

Header and footer

□ Header spans across all table columns

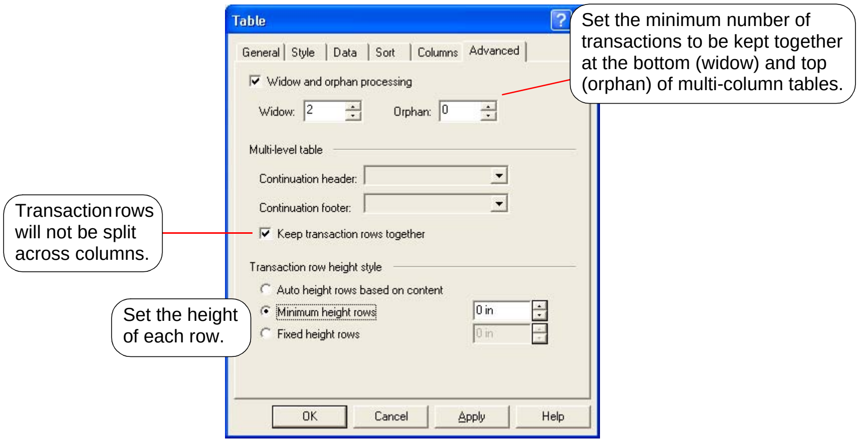
□ Footer spans across all table columns

Select the layout you want for the table on the page.

Choose the direction in which rows will be added.

Spread the header or footer across all table columns if required.

▶ **To change settings for transaction row:** click anywhere in the table and from the *Table* menu select *Properties/Table*. Choose settings from the Advanced page as required. Selecting *Keep transaction rows together* prevents transaction rows and any corresponding headers and footers from an individual repeating data element from being split across columns.



☑ Widow and orphan processing

Widow: 2 Orphan: 0

Multi-level table

Continuation header:

Continuation footer:

☑ Keep transaction rows together

Transaction row height style

○ Auto height rows based on content

● Minimum height rows 0 in

○ Fixed height rows 0 in

Set the minimum number of transactions to be kept together at the bottom (widow) and top (orphan) of multi-column tables.

Transaction rows will not be split across columns.

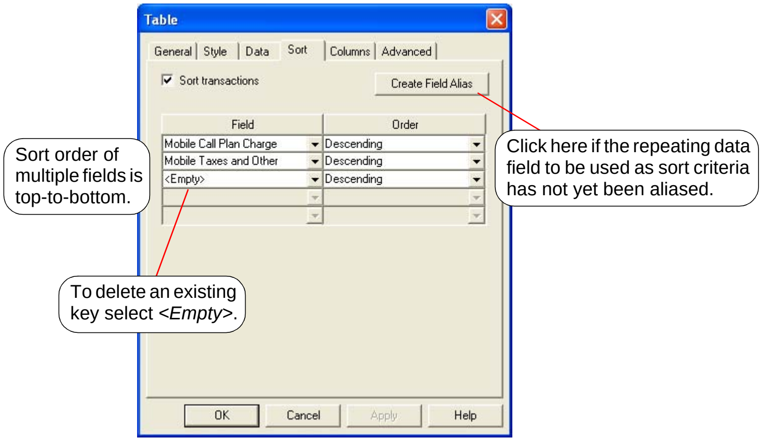
Set the height of each row.

Sorting

The repeating data to be placed within a transaction table can be sorted before being output. The contents of one or more fields within the repeating data are used as the sort keys. Data in each category can be sorted either in ascending or descending order.

> Sorting is based on the sequence of characters in the Unicode table. Results should be fully predictable when data contains standard ASCII-based characters but using sorting with other characters, particularly those from ideographic languages, may produce unexpected results.

▶ **To specify sorting options:** right-click within the table and select *Table Properties* from the shortcut menu. On the Sort page check the *Sort transactions* option. Enter the appropriate sort criteria using the dialog options.



☑ Sort transactions

Field	Order
Mobile Call Plan Charge	Descending
Mobile Taxes and Other	Descending
<Empty>	Descending

Sort order of multiple fields is top-to-bottom.

Click here if the repeating data field to be used as sort criteria has not yet been aliased.

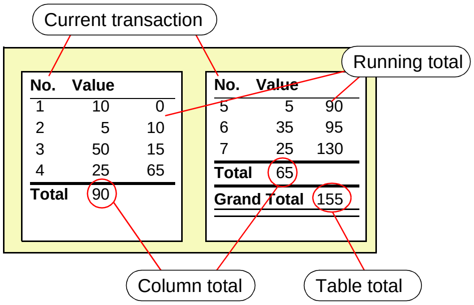
To delete an existing key select *<Empty>*.

Counts and totals

Entries in a transaction table are automatically counted and numeric values are dynamically totalled. You can use table functions to output the current count and total values within a table (or use them in table logic), as with any other value – see “Working with values” on page 104. The calculation of these counts and totals can be restricted to just the current column or page if required.

Transactions processed and the *Running totals* are only updated once each transaction has finished being processed, so if they are used within a transaction, the value will be that of the previous transaction. For instance, if you want to include a running total of a data field in each transaction, you must add the current data field to the running total.

Calculations with page and column scope can only be used in the *Table Header, Table Footer* or *After Last Occurrence* sections, where the final content of the page or column is known.



No.	Value	
1	10	0
2	5	10
3	50	15
4	25	65
Total	90	

No.	Value	
5	5	90
6	35	95
7	25	130
Total	65	
Grand Total	155	

Two columns of tables on a page showing counts and totals

▶ **To include the transaction counts and totals:** in the document view, position the pointer in the table where you want the count or total, and from the *Insert* menu select *Field/Table Function.*

Running total

Running total of specified value.

Transactions processed

The number of specified transactions processed up to that point, starting at 0.

Current transaction

The number of all transactions currently processed by table logic, starting at 1.

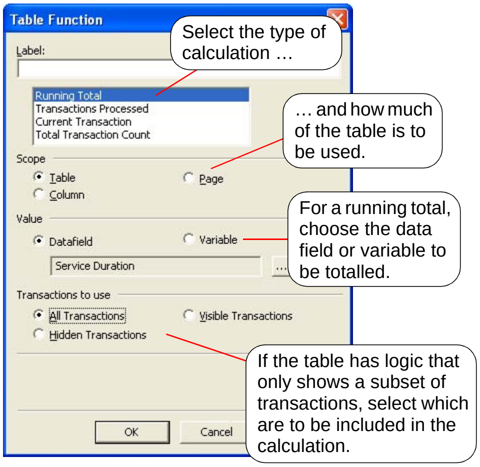


Table Function

Label:

Running Total
Transactions Processed
Current Transaction
Total Transaction Count

Scope
● Table ○ Page
○ Column

Value
● Datafield ○ Variable
Service Duration

Transactions to use
● All Transactions ○ Visible Transactions
○ Hidden Transactions

OK Cancel

Select the type of calculation …

… and how much of the table is to be used.

For a running total, choose the data field or variable to be totalled.

If the table has logic that only shows a subset of transactions, select which are to be included in the calculation.

Total transaction count

The total number of all transactions available to the table logic.

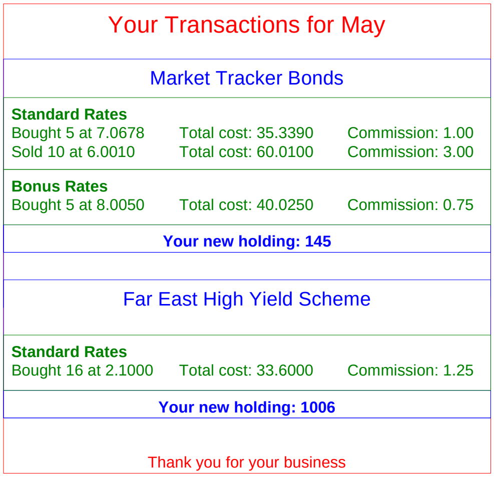
Storing counts for later use

The *Current transaction* and the *Total transaction count* can be saved in variables. This enables you to use this information elsewhere in the publication. These are only available for all transactions in the table, whether they are visible or hidden.

▶ **To store the transaction counts for later use:** right-click within the transaction table and select *Table Properties* from the shortcut menu. On the *Data* page, select the transaction counts you want to use elsewhere and specify the variable in which they are to be saved – see “Variables” on page 110.

Nested tables

While it is not possible to create a table within a cell of an existing table, it is possible to nest a table between the major row elements (in the *Detail* section) of another table. This may be useful if you need to present transaction data on multiple levels such as in the following example.



Your Transactions for May

Market Tracker Bonds

Standard Rates		
Bought 5 at 7.0678	Total cost: 35.3390	Commission: 1.00
Sold 10 at 6.0010	Total cost: 60.0100	Commission: 3.00
Bonus Rates		
Bought 5 at 8.0050	Total cost: 40.0250	Commission: 0.75

Your new holding: 145

Far East High Yield Scheme

Standard Rates		
Bought 16 at 2.1000	Total cost: 33.6000	Commission: 1.25

Your new holding: 1006

Thank you for your business

To do this you will need to use the logic map to mark the insertion point in the *Detail* section of an existing table when creating a second or third level table.

Nested tables can have headers and footers as with a regular transaction table. You can configure which of these should be shown if a new page is inserted while more than one level of table is being processed.

▶ **To create a nested table:**

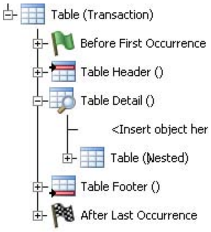


Table (Transaction)
Before First Occurrence
Table Header ()
Table Detail ()
<Insert object her
Table (Nested)
Table Footer ()
After Last Occurrence

1. Display the logic map if it is not already available.
2. Mark the insertion point within the existing table.

   Expand the table object and click on the *Table Detail* object.
3. Select *Table/Table Wizard* from the menu.
4. Use the table wizard to create the table settings in the normal manner.

▶ **To configure header and footer use:** right-click on a table object and select *Table Properties* from the shortcut menu. Use the *Advanced* page to specify *Continuation Header/Footer* settings. For instance, if *Show First* is selected only the header/footer from the highest level table will be output.

About design files and resources

At its simplest a DOC1 publication consists of a single private document design. In more sophisticated scenarios however, a publication can be made up of many different design files which are maintained independently and which can be shared by other publications. Each file is stored and controlled within the DOC1 repository.

The following types of design file can be created:

- **Publication** – an assembly of one or more related documents.
- **Document** – the basic design unit of a publication. Typically, publications have only a single document design which is made available automatically when the publication is created. However, you can create additional document designs for conditional selection or reuse within a range of similar publications.
- **Active Content** – a file containing a segment of document logic and/or presentation objects; an Active Content object is called as a function by a document.
- **External Documents** – pre-composed documents in a supported external format that can be inserted into publications.
- **Images** – graphic files that have been imported into the repository so they can be shared by multiple design files.

Additionally, design files may make use of **fonts** that have been imported from your output device environment or specifically shared. These are also stored as separate files within the DOC1 repository.

Before being published a publication must be linked with a **data format** that describes the structure and content of the input data file to be used in the production environment. Within document designs, references to input data are created as data record or data field objects that are aliases for the actual data elements that will be available in the input data file. Such aliases are stored for subsequent reuse within one or more **data dictionary** files. When publishing you will need to create or reference a **data map** that links the data aliases used in the design with the actual elements in the data format.

Depending on your working practices, you may decide to create some or all of these data related resources in advance of actually creating the document design. You can export a data dictionary and a data map file from a data format if required.

All design files and resources that are available for you to work with can be found and manipulated using the DOC1 Work Center Manager.

▶ **To show the Work Center Manager:** from the *View* menu select *Manager/All*.

For more information about:

Creating a basic document design see “Designing Page Layouts” on page 26.

Working with multiple documents design and external documents see “Publications and documents” on page 96.

Creating Active Content files see “Using Active Content” on page 224.

Importing font and image resources see “Using fonts and images” on page 85.

Adding data aliases to paragraphs and transaction tables see “Working with input data” on page 72.

Creating data formats, data maps and data dictionaries see “Identifying Input Data” on page 236.

Basic navigation of files in the repository see “Using the Work Center Manager” on page 21.

Advanced repository topics including the use of revision control and management tools see “Managing Resources” on page 373.

Version control and workflow

All design files within the repository are automatically version controlled.

When you create a new file or open an existing one a new **draft** version is created within the repository. While you are working with the file you can save it as many times as you need – each time the draft version is updated with your changes.

When the work is complete you should then **issue** the file. This marks it as the currently active version within the repository and makes it available for use with a publication when it is published.

▶ **To issue a design file:** click on it in a Work Center Manager view and select *Tasks/Issue* from the menu.

Only publications can actually be published. Other resources that are referenced by the publication are automatically built into the published version.

Publishing always uses the issued versions of publications and the other files they reference – i.e. draft versions of files will not be included. However, you can use the *Publish Current Draft* task to issue files automatically prior to publishing if required. Specifically, the last saved version of the publication is issued prior to publishing along with all the documents and other objects it references providing they are marked as private for the publication.

The *Publish Issued Version* task will ignore any draft versions of the publication or referenced resources that have not yet been issued.

ⓘ FOR DETAILS OF USING THE PUBLISH TASKS SEE “PUBLISHING AND DEPLOYMENT” ON PAGE 329.

Using fonts and images

Typically, standard Windows-based resources are used when designing publications. When creating text you can select a font from the list of resources available to the local computer. When including images you can select a graphics file directly from the local file system.

However, where it is important to design using resources only available in your existing output device environment you may need to import fonts, images into the repository. You may also want to add resources to the repository to make them sharable by other publications or users.

Fonts

When an existing publication is opened the Work Center searches the local machine to see if all the referenced fonts are available. If not, they are copied from the repository into the local windows font directory and automatically registered with the operating system.

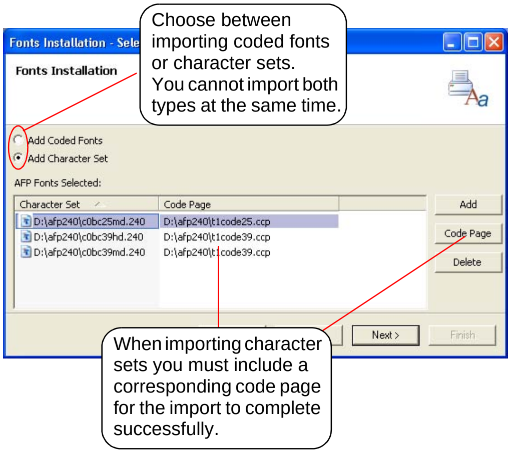
If you prefer, you can install all the fonts currently in your repository onto your local machine. This would be useful if, for instance, you have restored a different repository from backup.

Fonts imported into the repository are converted into Windows TrueType files but note that these will be bitmap rather than outline resources. This means that it is not possible to select a different style, size or color when working with an imported font. If you want to work with variations of such fonts you must import each variation required from a separate resource file.

> ⓘ FONT NAMES FOR AFP, XEROX AND IJPDS MUST NOT EXCEED 8 CHARACTERS IN LENGTH.

AFP

DOC1 Designer can import AFP raster type fonts (not vector or outline types). You may import single-byte AFP fonts either by reference to a coded font file or to a character set/code page pair. All single-byte fonts are assumed to be compatible with the IBM International code page 1148. Other single-byte code pages are not currently supported for font import. Coded font files contain references to the relevant character sets and code page files; these additional files are assumed to be present in the same directory as the coded font. DBCS AFP fonts can only be referenced by coded font name; you will need to specify explicitly the language encoding of the font to be imported.



Choose between importing coded fonts or character sets. You cannot import both types at the same time.

○ Add Coded Fonts
● Add Character Set

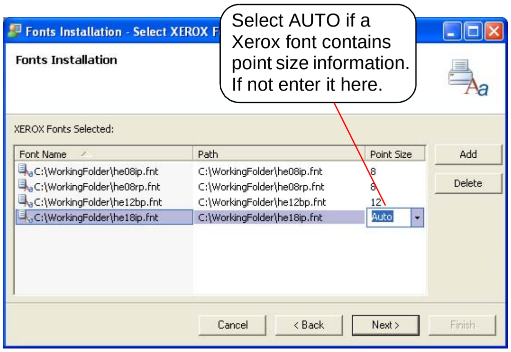
AFP Fonts Selected:

Character Set	Code Page
D:\afp240\c0bc25md.240	D:\afp240\t1code25.ccp
D:\afp240\c0bc39hd.240	D:\afp240\t1code39.ccp
D:\afp240\c0bc39md.240	D:\afp240\t1code39.ccp

When importing character sets you must include a corresponding code page for the import to complete successfully.

Xerox

DOC1 Designer can import the FNT file format used with Xerox Metacode printers. These are bitmap fonts with specific point sizes. In most cases the point size information is contained within the font itself but where this is not the case you may specify this as part of the import process. Note that many recent Xerox printers support PostScript printing for which Windows font resources are suitable.



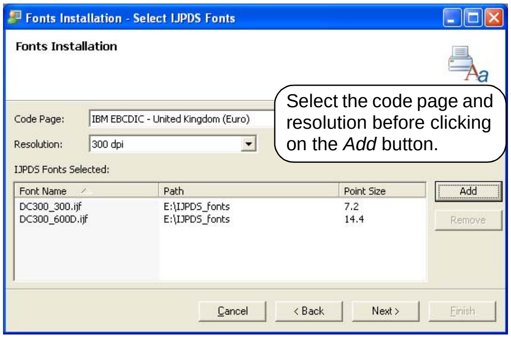
Select AUTO if a Xerox font contains point size information. If not enter it here.

XEROX Fonts Selected:

Font Name	Path	Point Size
C:\WorkingFolder\he08ip.fnt	C:\WorkingFolder\he08ip.fnt	8
C:\WorkingFolder\he08rp.fnt	C:\WorkingFolder\he08rp.fnt	8
C:\WorkingFolder\he12bp.fnt	C:\WorkingFolder\he12bp.fnt	12
C:\WorkingFolder\he18ip.fnt	C:\WorkingFolder\he18ip.fnt	Auto

IJPDS

DOC1 Designer supports a range of legacy fonts that may be present in some older IJPDS environments. Such fonts contain no information about the intended code page or resolution so you will need to specify these settings as part of the import function.



Select the code page and resolution before clicking on the *Add* button.

Code Page: IBM EBCDIC - United Kingdom (Euro)

Resolution: 300 dpi

IJPDS Fonts Selected:

Font Name	Path	Point Size
DC300_300.ijf	E:\IJPDS_fonts	7.2
DC300_600D.ijf	E:\IJPDS_fonts	14.4

Windows

Windows fonts are normally automatically added to the repository as you use them in publication designs but you may specifically import them in advance if required. This may be useful if you want to make specific fonts available to users in a shared development environment.

▶ To import printer fonts into the repository:

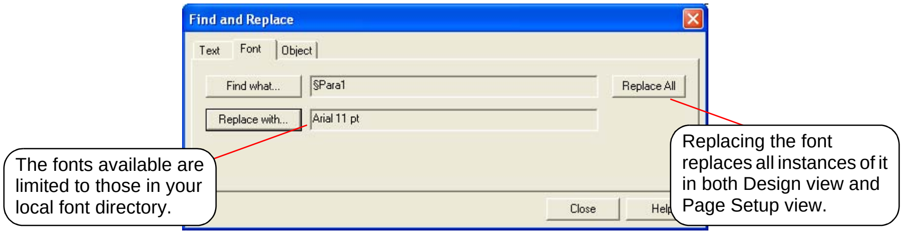
1. Select the *Environment* tab and then the *Fonts* bar
2. From the *Tasks* menu select *Import Fonts*.
3. Use the *Fonts Installation* dialog box to select the type of fonts to be imported and then identify the appropriate files.

▶ To install repository fonts to your local machine:

1. Select the *Environment* tab and then select the *Fonts* bar.
2. From the *Tasks* menu select *Install Fonts from Repository* and the dialog box is displayed.
3. Select the fonts you want to install from the list and click *Install*.

▶ To change fonts used in a document:

1. In the document editor select *Edit/Find and Replace* and the *Find and Replace* dialog box is displayed.
2. Click the *Font* tab and click *Find* what to display the *Fonts in use* dialog box.
3. Select *All replaceable fonts* from the drop-down list and select the appropriate font or character style for the find and replace action.
4. Click *Replace All* and all occurrences of the font or character style in the Design view and Page Setup view are replaced.



Find what... $Para1

Replace with... Arial 11 pt

The fonts available are limited to those in your local font directory.

Replacing the font replaces all instances of it in both Design view and Page Setup view.

Images and overlays

DOC1 can support a large range of image types including most common PC-based formats as well as those used with supported output devices.

When working with a design file you are able to reference images and overlays that are already stored within the Work Center repository or include new images that are available to the local file system. If you include a new image it will only be available to the current document unless you use the Make Public option.

ⓘ SEE “USING IMAGES” ON PAGE 51 FOR MORE INFORMATION.

Images and overlays can also be added directly to the repository using the Work Center Manager.

▶ To add an image to the repository:

1. Select the *Design* tab and then select the *Images* bar.
2. Select *Tasks/Import Images*
   – or –
   right-click in the navigator and click *Import Image* from the shortcut menu.
3. In the Open dialog box use the navigator to select the images you want to add and click Open.

Overlays

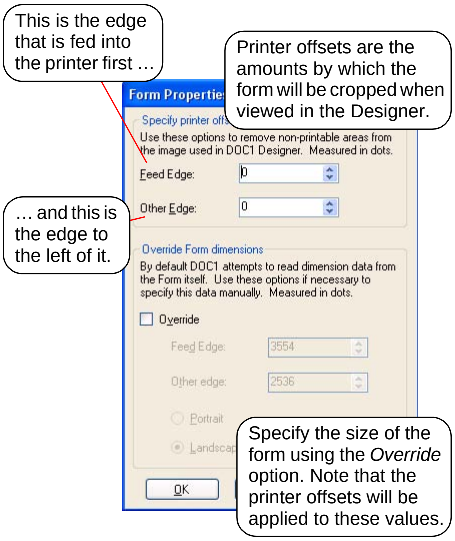
AFP overlays and Xerox Metacode forms typically contain pre-defined standard content such as corporate letter headings. They are often stored and used directly in the print environment and applied to output dynamically rather than being embedded in the output data stream itself. Where this is the case the overlay is simply referenced with the data stream. Where embedding is required DOC1 always passes an imported resource in its native format directly to the output data stream. In DOC1 Designer overlays can only be included as part of a document background. Note that AFP overlay names are restricted to a maximum of eight characters and Metacode form names to a maximum of six characters. See also “Overlays” on page 276.

Metacode forms sometimes do not include page size information. Where this is the case you will need to specify the relevant information when importing the resource so that DOC1 can correctly present an image of the form in Designer. Forms are also sometimes designed to take account of non-printable areas on the actual printer so you may need to specify printer offsets to compensate for this during import.

▶ To add an overlay to the repository:

1. Select the *Design* tab and then select the *Images* bar.
2. Select *Tasks/Import Overlays*.
3. In the *Open* dialog box use the navigator to select the overlays you want to add and click *Open*. Note that, whilst importing, all the resources – fonts, images, logos – must be available in the same location on disk as the overlay.

   For Metacode forms, the *Form Properties* dialog box will be shown for each overlay imported. Specify the properties as required. Metacode forms usually have the size of the form stored in the FRM. If it isn’t then you can specify it using the *Override* option.



Form Propertie

Specify printer offs

Use these options to remove non-printable areas from the image used in DOC1 Designer. Measured in dots.

Feed Edge: 0

Other Edge: 0

Override Form dimensions

By default DOC1 attempts to read dimension data from the Form itself. Use these options if necessary to specify this data manually. Measured in dots.

□ Override

Feed Edge: 3554

Other edge: 2536

○ Portrait

● Landscap

This is the edge that is fed into the printer first …

… and this is the edge to the left of it.

Printer offsets are the amounts by which the form will be cropped when viewed in the Designer.

Specify the size of the form using the *Override* option. Note that the printer offsets will be applied to these values.

Custom code pages

Custom code pages enable users to extend the number of code pages available in DOC1. By first exporting an existing code page and then modifying it in an appropriate text editor, you can then import it back into DOC1 as a custom code page. Currently DOC1 enables you to use custom code pages with data formats, lookup tables, AFP font generation and for viewing content in the DOC1 Editor.

This feature is aimed at DOC1 users who use extended character sets for double-byte and multi-byte languages, such as Chinese, Japanese and Korean. Languages that use single-byte code pages, such as English and German cannot make use of custom code pages.

As the large number of characters included in a code page makes the task of creating one from scratch unrealistic, it is recommended that existing system code pages are exported as templates for creating custom code pages.

Code pages are closely linked with both languages and subsequently fonts in DOC1. Using a custom code page with an unsuitable language or font can result in unexpected behavior. These issues normally arise if a double-byte or multi-byte code page is applied to a single-byte language or if a font associated with a single-byte language, such as Arial, is being used.

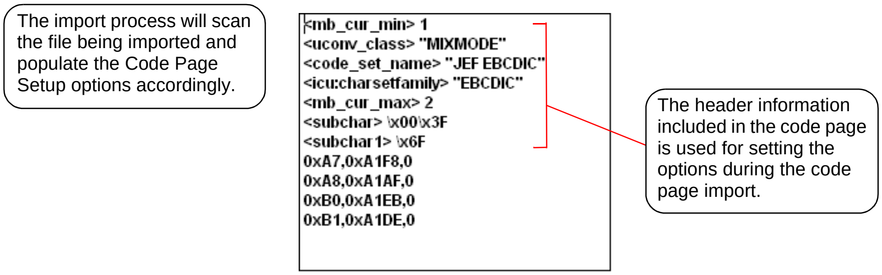
ⓘ DOC1's support of custom code pages is based on the IBM's open source ICU project. See http://www-306.ibm.com/software/globalization/icu/index.jsp for details.

Importing code pages

When creating custom code pages that will be imported into DOC1, they must be in either unicode mappings (ucm) or comma separated (csv) file format. You must ensure that any changes made to a code page outside DOC1 conforms to the expected syntax for the file, as there is no validation of the code page when it is imported into the Work Center.

▶ To import a code page:

1. Select the *Environment* tab and then select the *Other* bar.
2. Select *Tasks/Import code page*
   – or –
   right-click in the navigator and click *Import code page* from the shortcut menu.
3. In the *Import Files* dialog box navigate to the required ucm or csv file and click *OK*.
4. In the *Code Page Setup* dialog box apply the required settings for the code page and click *OK*.



```
<mb_cur_min> 1
<uconv_class> "MIXMODE"
<code_set_name> "JEF EBCDIC"
<icu:charsetfamily> "EBCDIC"
<mb_cur_max> 2
<subchar> \x00\x3F
<subchar1> \x6F
0xA7,0xA1F8,0
0xA8,0xA1AF,0
0xB0,0xA1EB,0
0xB1,0xA1DE,0
```

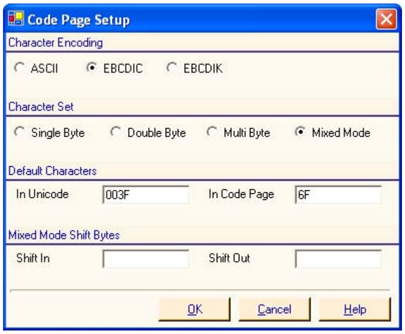
The import process will scan the file being imported and populate the Code Page Setup options accordingly.

The header information included in the code page is used for setting the options during the code page import.

Code Page Setup options

Character Encoding

DOC1 supports the following encodings: *ASCII*, *EBCDIC* and *EBCDIK* (an extension of EBCDIC that contains Katakana characters).



Code Page Setup

Character Encoding

○ ASCII ● EBCDIC ○ EBCDIK

Character Set

○ Single Byte ○ Double Byte ○ Multi Byte ● Mixed Mode

Default Characters

In Unicode: 003F In Code Page: 6F

Mixed Mode Shift Bytes

Shift In: Shift Out:

Character Set

The code page can be based on one of the following character sets:
Single byte – each character consists of a single byte.
Double byte – each character consists of two bytes.
Multi byte – each character can be made up of between one and six bytes.
Mixed mode – each character will be either single byte or double byte. The transition between the two is done using shift out and shift in characters.

Default Characters

These characters are used as replacement values when there are incomplete mappings between the DOC1 code page and the unicode values contained in the ucm or csv file. Typically the *In Unicode* and *In Code Page* characters used will substitute a question mark for missing mappings. This option is used when the *fallback indicator* calls for a substitution character.

Mixed Mode Shift Bytes

These options are only enabled if the *Mixed Mode* character set is selected. They indicate the start (*Shift Out*) and end (*Shift In*) of a group of double byte characters included amongst single byte characters.

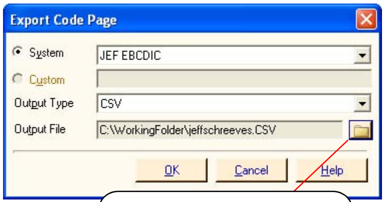
Exporting code pages

It is suggested that the system code pages included in DOC1 are used for creating custom code pages. To this end, you can export system code pages from DOC1. These can then be edited and imported back into the Work Center.

▶ To export a code page:

1. Select the *Environment* tab and then select the *Other* bar.
2. Select *Tasks/Export code page*
   – or –
   right-click in the navigator and click *Export code page* from the shortcut menu.

3. In the *Export Code Page* dialog box select a *System* or *Custom* code page, select a *CSV* or *UCM* output type and click the *Browse* button in the *Output File* field.
4. In the *Save Code Page* dialog box select a directory and enter a name for the code page and click *Save*.
5. Review the options in the *Export Code Page* dialog box and click *OK* to export the code page.



● System: JEF EBCDIC

○ Custom

Output Type: CSV

Output File: C:\WorkingFolder\jeffschreeves.CSV

Clicking the browse button displays the *Save Code Page* dialog box, enabling you to select the location for the code page.

Using the logic map

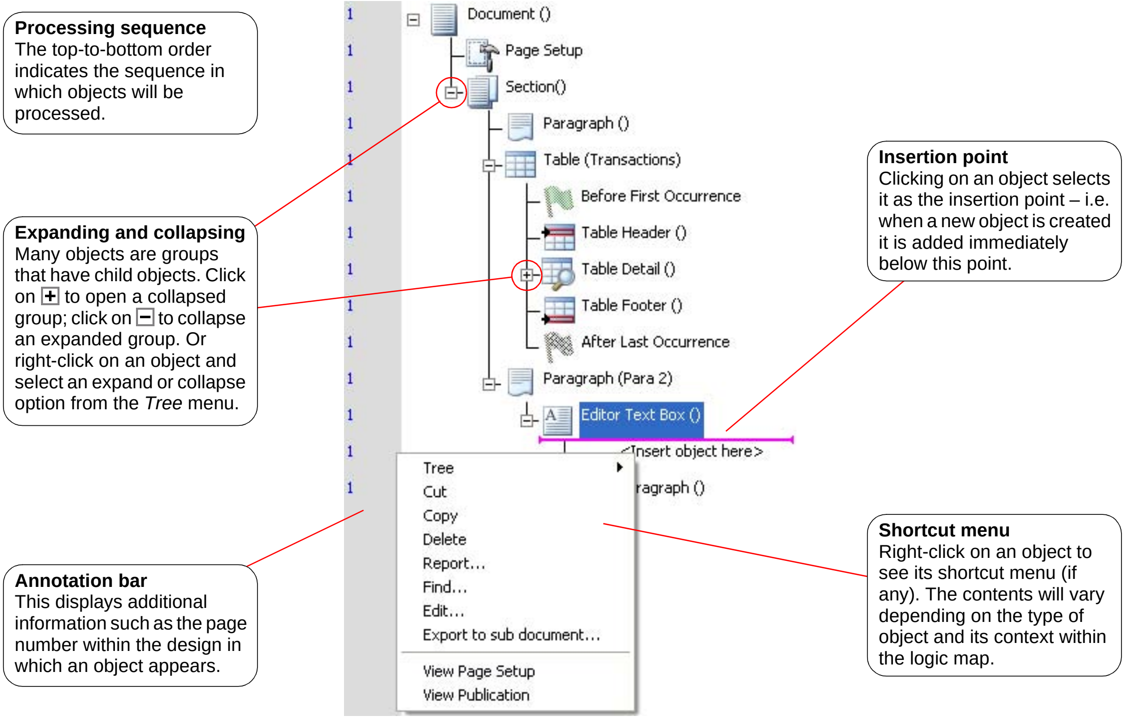
When you are editing design files the logic map view allows you to see a representation of the objects that make up the design and how they relate to one another. It is a visualization of the sequence in which the objects will be processed when the design is executed in the production environment.

The logic map is used to indicate where new objects are to be inserted. In order to work with variables, functions and other advanced features you will need the logic map to be available.

The logic map can also be used to move, copy/paste, and delete objects or groups of objects. Select an object and use standard keyboard controls or menu options to do so.

You can choose to deactivate a group of objects temporarily from the design using the Bundle feature.

▶ **To toggle the logic map view off or on:** select *View/Logic Map*.



Processing sequence
The top-to-bottom order indicates the sequence in which objects will be processed.

Expanding and collapsing
Many objects are groups that have child objects. Click on ⊞ to open a collapsed group; click on ⊟ to collapse an expanded group. Or right-click on an object and select an expand or collapse option from the *Tree* menu.

Annotation bar
This displays additional information such as the page number within the design in which an object appears.

Insertion point
Clicking on an object selects it as the insertion point – i.e. when a new object is created it is added immediately below this point.

Shortcut menu
Right-click on an object to see its shortcut menu (if any). The contents will vary depending on the type of object and its context within the logic map.

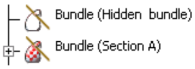
▶ **To move an object:** drag it to a new location using the mouse.
– or –
Use the *Cut/Copy/Paste* options from the shortcut menu.

▶ **To print part of the logic map:** right-click on a group object and on the shortcut menu click *Report*. All objects that form part of the tree for the group are printed.

Restricting active objects

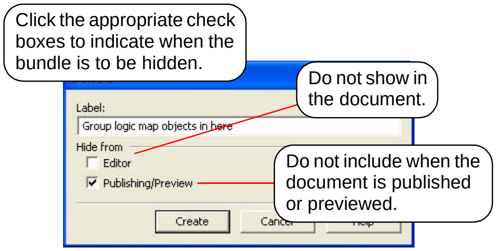
You can use the Bundle feature in the logic map to withdraw objects within a design file from being displayed or being processed. This can be a useful aid when testing, enabling you to concentrate on specific parts of the logic.

When hidden attributes have been specified, the bundle icon appears 'clear' in the logic map. If the contents are specified as hidden from the editor, only the bundle object itself is displayed in the logic map.



▶ **To create a bundle group:**

1. Mark the insertion point within the logic map.
2. Select *Insert/Tool Objects/Bundle* from the menu.
3. In the *Bundle* dialog specify the required 'hide' options and click *Create*.


Click the appropriate check boxes to indicate when the bundle is to be hidden.

Label: Group logic map objects in here

Hide from
☐ Editor
☑ Publishing/Preview

Do not show in the document.

Do not include when the document is published or previewed.

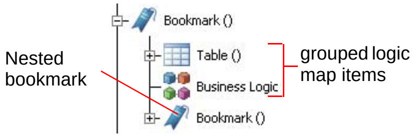
4. In the logic map move all objects to be hidden into the bundle group.

▶ **To reactivate hidden bundle objects:** double-click on the Bundle icon in the logic map and use the dialog to turn off the existing 'hide' options.

Bookmarks

You can provide bookmarks for applications that generate PDF output. As well as grouping logic map items within a bookmark object, you can also nest bookmarks to help create the required bookmark structure.

▶ **To create a bookmark:** mark the insertion point in the logic map and from the *Insert* menu select *Tool Objects/Bookmark*. Resolve the *Title* which provides the value for the bookmark.



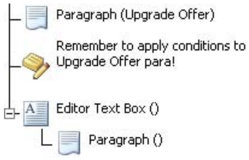
Nested bookmark

grouped logic map items

Notes

A note object allows you to provide a comment in the logic map.

▶ **To create a note:** mark the insertion point in the logic map and from the *Insert* menu select *Tool Objects/Note*. Enter the comment. The text does not automatically wrap so you must use carriage returns where required.

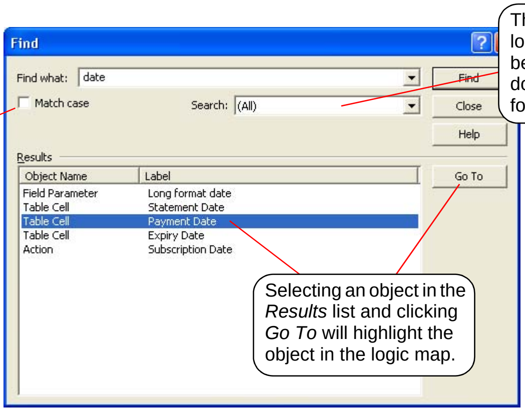


Searching the logic map

You can use the Find option to search the logic map. Find enables you to search for logic map objects by looking for the object or its label. You can search the entire logic map or narrow your search to specific objects.

▶ **To find a logic map object:** select *Edit/Find* and the *Find* dialog box is displayed. Enter the type of object or its label in the *Find what* field. Use *Match case* and the *Search* list as needed and click *Find*.

See also “Find and Replace” on page 31.



Find what: date

□ Match case

Search: (All)

Object Name	Label
Field Parameter	Long format date
Table Cell	Statement Date
Table Cell	Payment Date
Table Cell	Expiry Date
Action	Subscription Date

You can search for text that matches the case also.

The *Search* list includes the logic map objects currently being used in the document, enabling you to focus the search further.

Selecting an object in the *Results* list and clicking *Go To* will highlight the object in the logic map.

Logic map errors

From time to time your publication logic may contain inconsistencies that may prevent it being published. For instance, you may be supplying an object with a value that cannot be converted to the appropriate data type. Or you may have used a new variable within the design but never created logic to assign a value to it.

The Document, Publication and Active Content Editors evaluate your logic and provide information about this kind of problem within the Logic Status view. In the view you may click on a message to be shown the objects to which it relates (where applicable).

Attempting to insert objects in the logic map that cannot be placed at the selected insertion point causes the *Insertion Warning* dialog box to be displayed. This acts similar to the Logic Status view.

▶ **To view logic errors:** select *View/Logic Status* from the menu.

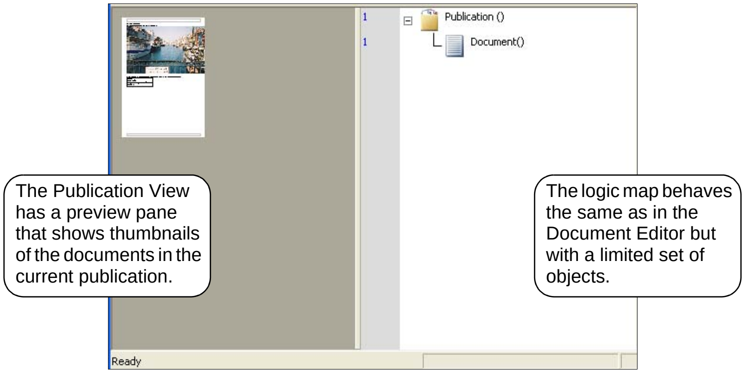
Publications and documents

Although many DOC1 applications consist of a single document design others often need to combine document types (both internal and external) into a single production job. For this reason all DOC1 application designs have a high-level Publication object that allows you to build such document combinations and to set global attributes for the entire job.

When you edit an existing publication the Document Editor will be shown first if it contains a single document. Otherwise the Publication Editor is loaded first.

When you create a new publication what you actually create is a new document design within a publication object. By default, you will be shown the Document Editor with the new document loaded. However, at any time you can switch to the Publication Editor that allows you to add further document designs required.

▶ **To switch to the Publication Editor:** click the *Publication Editor* toolbar button
– or –
right-click in the logic map of the Document Editor and from the shortcut menu select *View Publication*.



The Publication View has a preview pane that shows thumbnails of the documents in the current publication.

The logic map behaves the same as in the Document Editor but with a limited set of objects.

See “Creating conditional logic” on page 132 for information about creating page check, condition, case and loop objects. See “Production journals” on page 188 for information about using journals.

The Publication Editor has a logic map like the Document Editor but with a limited set of supported objects. This determines the sequence of documents within the publication and allows you to create conditional logic to decide when to use them if required. You may also add to a journal and use some action objects within publication logic.

A publication has a set of attributes that apply to all the documents it contains (although some may be overridden in other objects). In particular you can assign the sample data file used to provide example data in the document designs. Other optional attributes provide settings that will be used to configure document level information in the output datastream when the publication is used in the production environment. Note that any variables used to set attribute values must be specified in the publication logic map before the first document.

▶ **To set publication attributes:** double-click the Publication object in the Publication Editor logic map and the *Publication* dialog box is displayed.

Publication attributes

Job Options

Label

Identifies the publication in the logic map.

Data map

Allows you to assign a data map to the publication. This links data aliases used in document designs with actual data elements to be used in the production environment. See “Advanced data handling” on page 177 for details.

Sample data

Allows you to specify the sample data file that will be used to provide example data in document designs. See “Sample data and data mapping” on page 177 for more information.

Locale

Sets a default locale to be assigned to new documents added to the publication. Note that this is a default only and may be overridden for individual documents and sections within documents. For more information see “About locales” on page 183.

Spell Dictionary

Select a dictionary for your publication that can be used with the publication’s locale. Like the locale, this setting can be overridden for individual documents, sections and Active Content.

Automatically synchronize with referenced files

See “About synchronization” on page 101 for details.

Document information

These attributes allow you to set values that are required as document level settings in some output datastreams. They are ignored when creating datastreams other than those indicated although you may use them to provide reference information within your publication design as required. To enable these options select *Use Publication Attributes*.

Author

Select a value that identifies an author for each document. This setting is supported for PDF output.

Title

Select a value that provides a title for each document. This setting is supported for PDF and RTF output. This is also used as the <*Title*> element for HTML pages generated by the HTML for e-mail output device. See “HTML for e-mail” on page 308 for more details.

(continued)	**Publication attributes**
Subject	Select a value that provides a subject description for each document. This setting is supported for PDF output.
Keywords	Select a value that provides one or more keywords for document search functions when the document is opened in a browser. This setting is supported for PDF output.
Password	Select a value that provides a *user* password required to gain access to a document. This setting is supported for PDF output and will be automatically encrypted by DOC1 Generate when producing a PDF datastream.
Internet email address	Select a value that identifies the e-mail address to be associated with each document. This setting is supported for PDF output and will be used when the e-mailing function in the DOC1 DIME utility is used.
Internet email template	Select a value that acts as a string to be passed as a property into the e-mail. This string is then available for use by other software packages. This option is only enabled when the *Internet email address* option is checked.
Permissions	For PDF output the checked options indicate the permitted level of access to documents when opened by a recipient.

Style Settings

The *Style List* contains all the styles included in the repository that are available to the publication. These styles will be available from the *Styles* drop-down list when editing a document. You can add style names in the *Publication* dialog box, but the style's attributes can only be applied in the Document or Active Content editor. The *Remove* button can be used to remove the selected style from the list or permanently from the repository. For more information on styles, "Data driven character styles" on page 137.

Journal Options

This allows you to create and specify job-level journals such as start/end of job and DIJ. Journals that are to be generated within a publication are specified using the journal object in the logic map – see page 189 for details.

Create DOC1 Interchange Journal	Check this option if the output datastream created by this publication is to be stored in an e2 Vault or used by e-Messaging, and therefore requires a DOC1 Interchange Journal (DIJ) to act as the index of documents it contains. This will enable the *Edit* button where you can specify the details that will actually be included in the index. See "Interfacing with e2 Vault" on page 197 for more information.
Create Start of job journal	A journal will be created and records added once only, before any publications or documents are processed. If the production job consists of multiple publications, then all journals will be processed at the start. Use the *Edit* button to specify the contents – see page 189 for details.
Create End of job journal	As above but journals will be generated after the last publication is processed.

Using multiple documents

ⓘ For more information see “About DOC1 input data” on page 237

By default, a publication is made up of a design for a single type of document. However, you can add as many document designs as required to a publication providing any input data they use can be provided within the same publication data set. A publication data set consists of the data between records defined as the *Start of publication* in a data format.

You may want to break down your publication into multiple documents if, for example you

- have document designs you want to share with other publications, see “Sharing and synchronization” on page 101 for details.
- have different document designs for a range of output devices.
- want to select document designs conditionally.
- want to use the same publication data for different document designs.

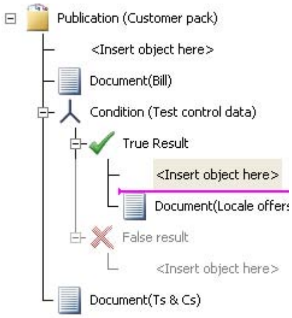
There are two different methods for including documents conditionally within a publication. One method is to use a condition within the publication logic map. For more details see “Condition” on page 133. You can also use the *Show When* option for specific documents. When this type of condition is placed within a document you can include it in multiple publications without having to recreate the condition each time it is used.

By default, all documents will be included in all the output datastreams produced by a publication. If you have document designs that are only suitable for specific output methods you can explicitly assign the appropriate types of output datastream to the document. In the production environment DOC1 Generate will only use the document design when it is writing to suitable output datastreams.

▶ To create a new document within a publication:

1. Switch to the Publication Editor.

   Click the *Publication View* toolbar button
   – or –
   right-click in the logic map and from the shortcut menu select *View Publication*.



2. Adjust publication logic if required.

   In the Publication Editor logic map you can create any new conditional logic that may be required to select the document for processing.

3. Add a document

   In the logic map mark the insertion point as required. From the *Insert* menu select *Document/New.*

4. Specify document attributes

   Double-click the new document icon in the logic map to display its attributes.

   Assign a label and use the *Locale* field if you want to override the default locale for the new document. If you choose a new locale you should also select a corresponding dictionary for the spell checker. Where you intend to use paragraph localization you should also assign the languages for which paragraph variants are required using the *Languages* tab.
   See “Localization” on page 183 for information about locales and paragraph localization.

   Check the *Continue page count* option if you want the Document Page Number system variable to retain the number of pages generated by the previous document in the publication (if any). See “System variables” on page 155 for more information.

   If appropriate select one or more output datastreams for the document from the *Supported Output Streams* tab. Note that the default output setting for this is *All*.

5. Edit the new document

   Right-click the document icon in the logic map and select *View Document* . The Document Editor is invoked with the document loaded.

▶ **To make a document conditional in a single publication:** in the Publication Editor logic map insert a Condition from the *Control Object* menu

▶ **To make a document conditional in all publications in which it is used:** in the Publication Editor double-click the document in the logic map and the *Document* dialog box is displayed. Click the *Show When* tab, select the *Show When* check box and enter the required condition. For details on creating conditions, see “Conditional expressions” on page 132.

▶ **To use an external document in your publication:** in the Publication editor double-click on the relevant document and in the *Document* dialog box *External Documents* page, browse for the file you want to import, and click *Update*. Each page of the imported document will be placed behind everything else on the pages in your publication. If necessary, additional pages will be created to accommodate each imported page.

▶ **To update an external document:** import the updated external document (see above) ensuring that the name is the same. All publications that use this external document will then use the updated document.

▶ **To remove an external document from the publication:** double-click on the relevant document and in the *Document* dialog box *External Documents* page, click the *Delete* button.

Sharing and synchronization

You can reuse parts of your publication design by creating Active Content files that contain commonly used sections of logic and presentation objects. Once you have issued a Active Content file it can be included in your publication or document design.

ⓘ For more information see “Using Active Content” on page 224

You are also able to share complete document designs among multiple publications. By default however, documents are solely available for use with the publication in which they are created and you will need specifically to make them *public* before they can be reused.

Documents included in a publication will always inherit data assigned to the publication. When editing *public* documents you can associate data map and sample data independently of the publication if required.

▶ **To make a document public:** click on the appropriate document icon in the publication editor and select *Edit/Document/Make Public* from the menu.

▶ **To add an existing public document to a publication:** In the publication editor mark the insertion point in the logic map. From the *Insert* menu select *Document/Existing*. Use the dialog box to select from the documents already in issue. Remember that all documents in the same publication must share the same input data file and publication data set definition.

▶ **To edit a public document from a parent publication:** in the publication editor select and right-click on the required public document in the logic map and choose the *Open option* from the shortcut menu.
– or –
open the public document in the document editor, right-click the *Read Only* document object in the logic map and select *Open* from the shortcut menu.

About synchronization

Active Content and public documents can be edited either independently by launching them from the Work Center Manager or directly within the context of a parent document or publication that makes use of them.

If you choose to launch a shared file from a parent editor certain attributes from the parent are automatically passed to the new editor session:

- Data map and sample data assignments.
- Variable assignments.
- Locale settings.
- Highlight color definitions.

When working in this way you may want to use the *Synchronize with open files* option to refresh this exchange of data to reflect the current state of associated files open in other editor sessions. If you choose to synchronize with open files from a child editor session the parent is updated automatically. You need to be aware that changes are simply made available to other editor sessions containing related information and must be synchronized on an individual basis to view changes.

ⓘ Note that if you are editing Active Content in a conditional path that is not currently active, variables, locale and highlight color assignments may not be the most recent.

▶ **To synchronize the current editor with open files:** from the *View* menu select *Synchronize Editor/With Open files*.

If you launch a file independently the editor does not, by default, have access to such ‘shared’ values that may be provided or updated by other files that it references. For example, if a document includes an Active Content file that sets values for its parent you may often see <Not Set> when such values are displayed. You can use the *Synchronize with referenced files* option to access such lower level logic and update the display to take account of it.

▶ **To synchronize the current editor with referenced files:** click the *Synchronize with referenced files* toolbar button
– or –
from the *View* menu select *Synchronize Editor/With Referenced files*.

If required you can make this type of synchronization the default behavior for a publication by choosing the *Automatically synchronize with referenced files* attribute. Note, however, that for complex applications and those using many referenced files this may significantly affect the load time of the publication. See “Publications and documents” on page 96 for details of setting publication attributes.

External documents

In addition to building a publication using multiple DOC1 document designs you can also import PDF and Quark generated EPS files into the Work Center and include them as pre-composed documents within a publication.

Once included in a publication, the complete external document is viewable from the Work Center Manager.

PDF DOCUMENTS THAT ARE PASSWORD PROTECTED CANNOT BE IMPORTED.
ONLY QUARK GENERATED EPS FILES CAN BE USED AS AN EXTERNAL DOCUMENT, OTHER EPS FILES ARE CURRENTLY UNSUPPORTED.
NOTE THAT THE ABILITY TO IMPORT EXTERNAL DOCUMENTS IS INDICATED BY YOUR DOC1 LICENSE.

The external document itself is treated as a single object, similar to an image, regardless of its size or content. It is placed behind the page content, including any background and each page is aligned from the top left corner. If necessary, extra pages will be created in the publication to contain each imported page.
Note that you cannot change the actual content of the imported external document although you can re-import an updated document.

▶ To import a PDF or EPS document into the Work Center:

1. In the Work Center Manager select the *Design* tab and click the *External Documents* bar. In the *Tasks* menu, select *Import External Documents*.

2. Browse to the required file and click *Open*. The external document is imported into the Work Center and is ready to be used in a publication.

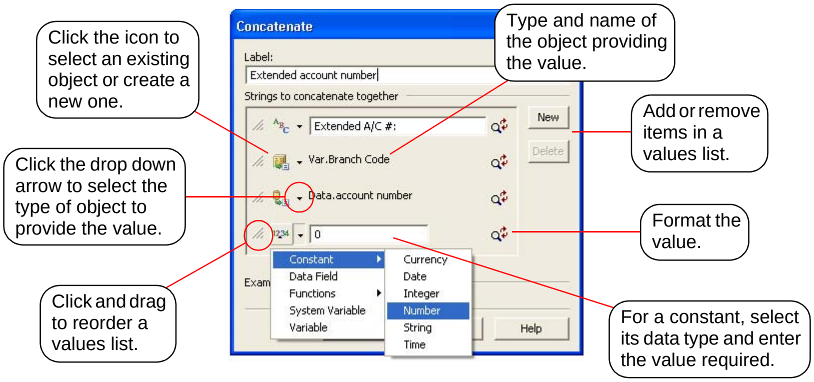
▶ To browse an external document: in the Work Center Manager select the *Design* tab, click the *External Documents* bar and open the relevant document.

Working with values

Adding values to objects

Many objects require one or more values to be provided as input to their function. For instance, a table lookup requires a text parameter; a condition needs two values of the same type to compare; and a substring requires the text to be manipulated plus offset and length parameters.

Values can be provided by data fields, variables, system variables, the output from functions or you can simply enter a constant. You can also use formula expressions that allow you to create more complex parameters or as a more convenient way of adding compound functions. See “Appendix B – Formula expressions” on page 414 for details. You identify the item that will provide the value (or the value itself in the case of constants) as part of the appropriate object dialog box.



Concatenate

Label: Extended account number

Strings to concatenate together

Extended A/C #:

Var.Branch Code

Data.account number

0

Click the icon to select an existing object or create a new one.

Click the drop down arrow to select the type of object to provide the value.

Click and drag to reorder a values list.

Type and name of the object providing the value.

Add or remove items in a values list.

Format the value.

For a constant, select its data type and enter the value required.

About data types

All values that are stored or output by a publication have assigned data types that tell DOC1 how to handle their contents. Such data types may be explicitly or implicitly defined depending on the object providing the value and the circumstances in which it is used. For example, when you create a new variable or data field you must explicitly specify the type of data it will hold but when you use the concatenation function its output is implicitly a string.

DOC1 supports the following data types:

String: a string of characters or RTF with basic text formatting. String values provided by a data field have a limit of 32k characters.

Number: a decimal number with a maximum of 24 digits before the decimal place and 8 after.

Integer: a whole number up to 9 digits long.

Currency: numeric data representing monetary values with a maximum of 24 digits before the decimal place and 8 after.

Date: a calendar date.

Time: a time of day value.

Some types of data may be temporarily converted at the point where they are used to provide a different type. For example, a number can be adjusted to a string for use with text related functions.

ⓘ CALCULATIONS THAT RESULT IN NUMBER VALUES WITH WHOLE OR FRACTIONAL PARTS LONGER THAN THE SUPPORTED MAXIMUM WILL RESULT IN DATA BEING TRUNCATED. FOR INSTANCE, THE CALCULATION 0.00009 X 0.0001 WILL PRODUCE THE RESULT 0.00000000 AS THE ACCURATE RESULT OF 0.000000009 IS TOO LONG FOR THE FRACTIONAL PART OF THE NUMBER DATA TYPE.

Formatting values

All objects that provide a value have a default presentation format that is assumed from their data type, the circumstances in which they are used and the DOC1 locale that is active when the object is processed. For example, if a locale based on regional settings for the United States is active a value of type *Date* will typically be assumed to have the format mm/dd/yyyy. Similarly, a value of type *Number* will have a default output that includes two decimal places using the period character as the decimal point and will always be assumed to be a decimal value unless specifically cast to type *Integer*.

An initial locale is specified for each publication. This can be altered during processing by using a *Change locale* action.

You can override the default formatting when an object value is:

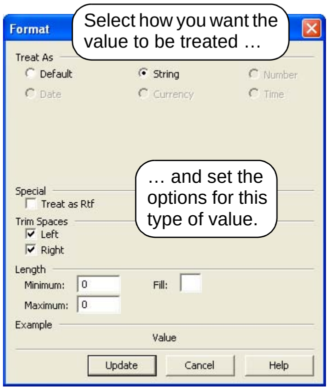
- included in a presentation object
- used with a function requiring string input
- used to update a variable of a dissimilar type.

> As well as setting defaults for design objects, DOC1 locales are also used to specify custom or variable formatting of actual document content to support regional differences. For instance, you can specify the names to be used for days of the week or months, or the character to be used for the decimal point. Different locales can be activated during a production run so that these preferences are used to format output.
>
> See "About locales" on page 183 for details of creating and editing a locale.

Where mixed data types are used as part of the same object DOC1 will attempt to convert (cast) values automatically as appropriate. For instance a string value of 15 August 2005 can be used to update a date variable. When there is ambiguity between data types used in functions or conditional comparisons you are also able to cast values explicitly to another data type.

Where values cannot be converted (either automatically or explicitly) an error will occur.

▶ **To specify formatting of a data field or variable placed on a page:** right-click on the field in the document editor and select *Format* from the shortcut menu
– or –
double-click the corresponding *Field Parameter* object in the logic map.
Click the format button and select the required options in the *Format* dialog box.



Select how you want the value to be treated …

… and set the options for this type of value.

Format

Treat As
○ Default ● String ○ Number
○ Date ○ Currency ○ Time

Special
□ Treat as Rtf

Trim Spaces
☑ Left
☑ Right

Length
Minimum: 0 Fill:
Maximum: 0

Example
Value

▶ **To specify formatting & casting of variables used within functions or objects:** in the appropriate object dialog box click the format button adjacent to the value to be formatted. Select the required options in the *Format* or *Casting* dialog box.

Format dialog box

String options

Special

Use the *Treat as RTF* option to identify a string value as being in RTF format. This can be either embedded *RTF code* or a reference to an *external RTF filename*. See “RTF fields” on page 108 for more details.

Trim spaces

If selected, any spaces to the *Left* or *Right* of other field content are removed before the field is placed. This is not available for RTF fields.

Length

If the number of characters in the field is less than the *Minimum length* the remaining positions will be padded with the *Fill character*. The space character is used for this purpose unless specified otherwise. This is not available for RTF fields.

Number options

Format

The *Negative* and *Positive* formats define optional text strings to be appended to number fields when the value is less than zero or greater than zero respectively. You can select from the list of standard formats or enter your own string.
The *Zero as* options allow you either not to display values where the value is zero or to display a string of up to 10 characters in its place. For instance, you may want to show the word 'Nil' rather than 0.

Decimal

Decimal places indicates a fixed number of decimal places to be shown by this field. Positions not supplied by input data are padded with zeros.
If *Exclude decimal point* is checked decimal points in the field value will be ignored. For example, if the field contains 120.07 it will be formatted as 12007.
If *Include thousands* separator is checked a separator character will be used to group digits in number values in excess of 1000. For example, assuming the separator character is a comma 2108910 becomes 2,108,910.
If *Round to last decimal* is checked the last digit in the number will be rounded up or down to represent any remaining fraction when the number of decimal places in the value exceeds that specified. If this is not checked the number will simply be truncated to the specified number of places. For example, assuming 2 decimal places are specified, 100.009 becomes 100.01 if rounding is checked otherwise it becomes 100.00.

Length

As for *String* formats.

Date options

Default Short

Uses the short date format of the active locale.

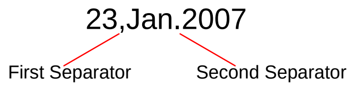
Default Long

Uses the long date format of the active locale.

Use location defaults

Use these buttons to set all *Date format* options to the styles defined for either long or short dates in the active locale.

(continued)	**Format dialog box**
Order	Specifies the order in which the three components of a date field will be presented. Note that the use of some options will mean that one or more components will not be used.
Day, Month and Year	Specifies the style to be used to represent each of the date components.
First and Second separator	Specifies the characters that will used to separate the three components of a data field.



Currency options

Options are identical to Number formats with the exception of:

Currency symbol	Select a character to be used as a prefix for the value.

RTF fields

You can dynamically include content in Rich Text Format (RTF) by passing it within a value object such as a data field or a return from a lookup table. Such objects can either contain actual RTF code or be a reference to an external file that contains RTF. RTF code would normally be included using RTF input data – see “Including RTF fields in input data” on page 182. Note that RTF fields are only supported on Windows and Unix platforms.

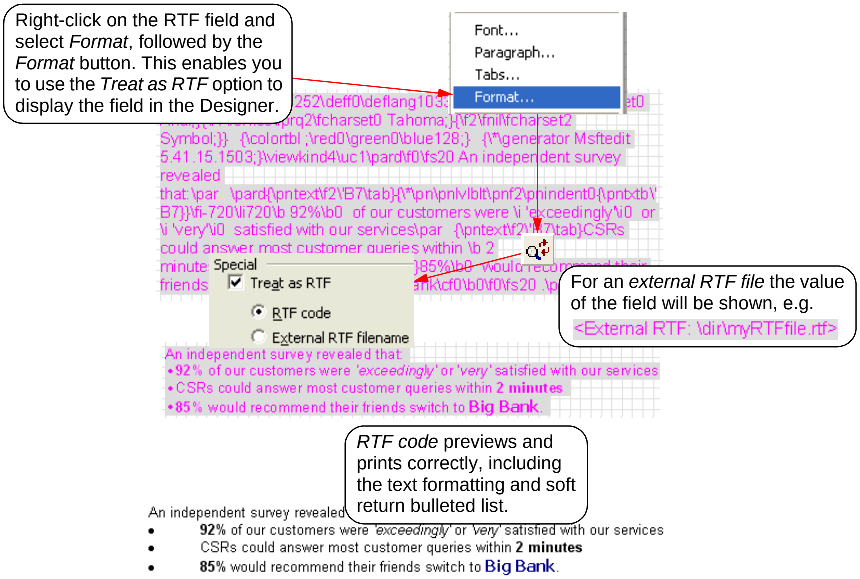
You can also import the contents of a complete RTF file directly into your document. See “Importing paragraph content” on page 44.

In the production environment DOC1 will format the RTF as a single uninterrupted piece of content at the point at which it is placed on the page by the publication logic. The RTF content will automatically flow onto new pages as appropriate. It is the user’s responsibility to make sure that such content is only placed where it can be handled correctly; for example, RTF placed in a text box or other container may be truncated.
Note that the Preview feature will only be able to include content from a referenced external RTF file if the reference is to a file that is available to the DOC1 Designer.

Simple text formatting, such as bullets, indents and font decoration can be used in the RTF field – see “RTF” on page 299 for a list of restrictions. The fonts used by the RTF field need to be available to the Work Center, otherwise the closest matching fonts from the DOC1 repository will be used. Nested bullets, tables, boxes, lines and images are not supported. Hard returns can only be used when the RTF is supplied as XML input data.

You can use variables within an RTF data field using the *DocVariable* field option in Microsoft Word. When you create the RTF in Word you can include a reference, by name, to a variable contained in a DOC1 application. Once the variable has been set in your design document it will be displayed when the RTF field is inserted. The variable in DOC1 must be of type *string* and *document* scope. For more details on using Word's field option consult your Microsoft documentation.

When an RTF field is placed in a document it is displayed in RTF code. The field must be designated as RTF for it to be presented correctly in the Designer and processed accordingly by DOC1GEN. See "Formatting values" on page 106 for more details.



```
252\deff0\deflang103                                          et0
prq2\fcharset0 Tahoma;}{\f2\fnil\fcharset2
Symbol;}}  {\colortbl ;\red0\green0\blue128;}  {\*\generator Msftedit
5.41.15.1503;}\viewkind4\uc1\pard\f0\fs20 An independent survey
revealed
that:\par  \pard{\pntext\f2\'B7\tab}{\*\pn\pnlvlblt\pnf2\pnindent0{\pntxtb\'
B7}}\fi-720\li720\b 92%\b0  of our customers were \i 'exceedingly'\i0  or
\i 'very'\i0  satisfied with our services\par  {\pntext\f2\'B7\tab}CSRs
could answer most customer queries within \b 2
minute                       }85%\b0  would recommend their
friends                      ank\cf0\b0\f0\fs20 .\p
```

Special

☑ Treat as RTF

● RTF code

○ External RTF filename

An independent survey revealed that:
- **92%** of our customers were *'exceedingly'* or *'very'* satisfied with our services
- CSRs could answer most customer queries within **2 minutes**
- **85%** would recommend their friends switch to **Big Bank**.

Right-click on the RTF field and select *Format*, followed by the *Format* button. This enables you to use the *Treat as RTF* option to display the field in the Designer.

For an *external RTF file* the value of the field will be shown, e.g. <External RTF: \dir\myRTFfile.rtf>

RTF code previews and prints correctly, including the text formatting and soft return bulleted list.

An independent survey revealed that:
- **92%** of our customers were *'exceedingly'* or *'very'* satisfied with our services
- CSRs could answer most customer queries within **2 minutes**
- **85%** would recommend their friends switch to **Big Bank**.

Casting

In most circumstances the data type required by a particular object is implicit and if you specify a value of a different type it will, where possible, automatically be converted to the required type. For example, if you are storing a number in a currency variable, the number would be converted to currency automatically.

In some situations however, the expected data type can be ambiguous, particularly when working with mixed type values as part of arithmetic or conditional functions. In these circumstances you should explicitly cast the values to the required type.

Where appropriate the *Casting* dialog box is automatically invoked in place of the *Format* dialog box when you select the format button within object dialog boxes.

Note that casting provides a single conversion of a value as input for the object for which it is specified. It does not influence the original object value which retains its original data type.

Variables

Variables store values for subsequent use and can be updated as required. You can use a variable to provide input to an object or output the value it holds within a paragraph.

For example, you may want to use a variable to add number values from different sources before presenting the total, or to evaluate and hold the most recent date from a range of date fields.

A variable is an array which can be used to store multiple values of the same type. Different elements within the array are referenced by an index. The index is identified by an integer from 1 to n. The index can be provided by a value object, such as an arithmetic function or a data field.

ⓘ TO USE A VARIABLE TO PROVIDE THE INDEX FOR AN ARRAY ITS SCOPE CANNOT BE LOCAL. IF THE SCOPE OF A VARIABLE IS CHANGED TO LOCAL IT INVALIDATES ANY INDEX VALUE.

Variables can hold values of any of the supported data types but each individual variable can only accept data of a single type. The data type expected by a particular variable cannot change within the same publication.

By default, the value stored in a variable is cleared at the start of each new publication. Users with the appropriate design privileges can, if required, either extend this scope to an entire job or limit it to individual documents, sections or Active Content.

New variables can be created as they are used or set. When inserting a variable into a document, you must ensure that, in the logic map, the variable is assigned the required value before the paragraph object in which it is included. Until it has a value, it will be displayed as <Not Set>. Variables that have an empty value are displayed as <Empty> when the formatting marks option is used.

▶ To insert a variable in a document:

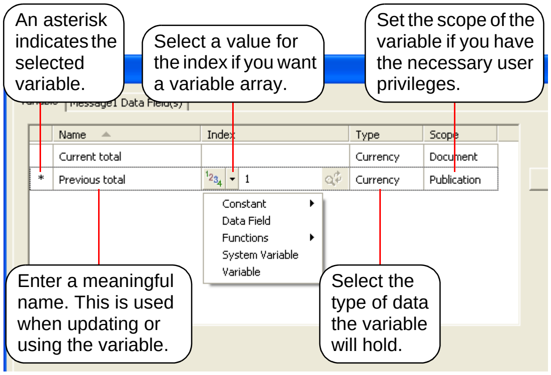
1. In the document view place the cursor in the relevant paragraph.
2. From the *Insert* menu select *Field/Variable*. The *Variable* dialog box is displayed.
3. In the *Variable* dialog box either select an existing variable or create a new one.

   If the variable is an array click in the index field and select a value for the index number.
4. Click the *Update* button.

   If you want to format the variable, see “Formatting values” on page 106 for details of the options available.

▶ To use variable contents within another object: see “Adding values to objects” on page 104.

▶ **To create a new variable:** in the *Variable* dialog box click the *New* button and enter a new name in the empty entry. Select an index entry if the variable is to be an array. Specify the attributes of the variable and click *Update*. You can then assign a value to the variable.



	Name	Index	Type	Scope
	Current total		Currency	Document
*	Previous total	1	Currency	Publication

An asterisk indicates the selected variable.

Select a value for the index if you want a variable array.

Set the scope of the variable if you have the necessary user privileges.

Enter a meaningful name. This is used when updating or using the variable.

Select the type of data the variable will hold.

Scope options

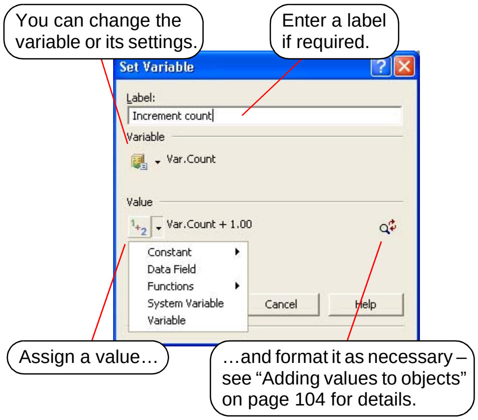
Job The value held in the variable is never cleared. You can use variables with Job scope to provide counters or totals across an entire production run for example.

Publication The value held in the variable is cleared at the start of each new publication.

Document The value held in the variable is cleared at the start of each new document within a publication. Use this setting if you need to reuse the same logic across multiple document designs or to use the same variable name for different purposes in other documents within the publication.

Local The value held in the variable is cleared at the start of each new section or Active Content.

▶ To assign a value to a variable:



You can change the variable or its settings.

Enter a label if required.

Set Variable

Label: Increment count

Variable: Var.Count

Value: Var.Count + 1.00

Assign a value…

…and format it as necessary – see “Adding values to objects” on page 104 for details.

1. Mark the insertion point in the logic map where you want to set the value of the variable.

2. From the *Insert* menu select *Set Variable*. Choose or create a variable in the *Variable* dialog box and select *Update*. The *Set Variable* dialog box is displayed. Provide a label and set the value for the variable and click *Create*.

   You may often want to use a function such as arithmetic or concatenation to add to existing contents rather than just replacing them. See “Adding values to objects” on page 104 for details.

   Note that assigning a value actually creates a *Set Variable Value Action* object in the logic map.

Deleting variables

Variables that are not used are automatically deleted when the document is closed. Only the variables that are used are saved. As such, you cannot specifically delete a variable. If you do want to delete one, you must ensure that it is not used by any document or publication. It will then be deleted automatically on closing.

Page and job handling

Generating new pages

New pages are automatically created when the body area of the current page is full. However you can also use the logic map to insert an explicit new page at any point.

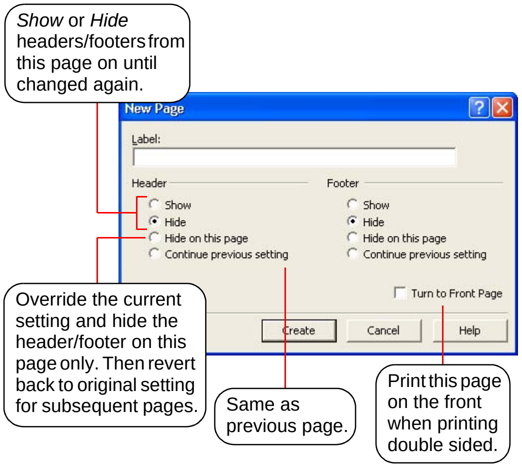
▶ **To insert a new page:**

1. Mark the insertion point in the document logic.

   The new page is actioned as soon as the object is encountered.

2. On the *Insert* menu click *Control Object/New Page*.

3. Use the *New Page* dialog box to set header and footer options and the duplexing setting.
   See also “Pagination” on page 43.



New Page

Label:

Header
○ Show
● Hide
○ Hide on this page
○ Continue previous setting

Footer
○ Show
● Hide
○ Hide on this page
○ Continue previous setting

□ Turn to Front Page

Show or *Hide* headers/footers from this page on until changed again.

Override the current setting and hide the header/footer on this page only. Then revert back to original setting for subsequent pages.

Same as previous page.

Print this page on the front when printing double sided.

Page Setup view

The Page Setup view contains logic that is performed every time a document generates a new page. Page Setup has a range of logic map groups that are executed in a fixed sequence for each page generated by a document. The table below identifies these groups and what they might be used for.

▶ **To work with the Page Setup view:** click the toolbar button or right-click in the logic map and select *View Page Setup* from the shortcut menu.

Headers and footers

The Default Header and Default Footer objects are used to define the height of headers and footers in the current document and to format how these areas appear on the page. You can conditionally adjust the height of the header/footer areas within the Default logic. These settings apply to the entire document and cannot be adjusted in any other logic map groups.

Header Content and Footer Content objects are used to add content to header and footer areas. These can be created anywhere in the page setup view and are often used within conditions where selective content is required. For instance, you may want to include a different header/footer style on odd and even pages.

Note that only one header or footer is active at a time. If more than one Header Content or Page Content object is encountered while processing the page setup logic only the content of the last object processed will be used.

You can also include content for the header/footer with the Default Header and Default Footer groups but note that this will be replaced if Header Content or Footer Content objects are processed elsewhere in the page setup logic.

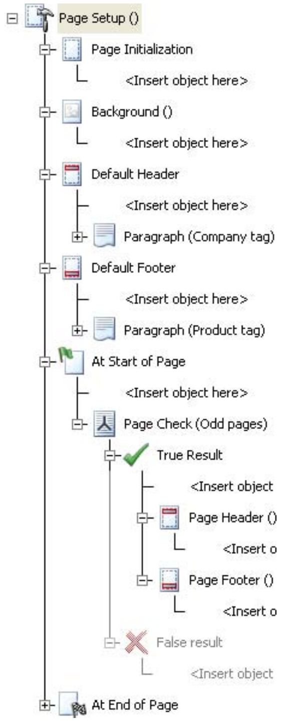
Headers and footers can be turned on or off individually, effective for the entire document. The page content will then use any additional space. Note that the default is for headers and footers to be on.

ⓘ Where the Page Setup view is used without the logic map being available all presentation objects are automatically added to the Default Header and default Footer groups.

▶ **To add basic header or footer content:** click inside the header or footer area and create paragraphs and graphics as normal. The objects are added to the Default Header/Footer group.

Page Setup group	Logic is executed	Can be used for
Background	Once the page initialization logic is complete.	Adding content to the background layer for a watermark effect or similar (see “Background” on page 62). Creating reserved areas that are unavailable to the main document body. Creating margin note areas for placing feature content in a margin note, positioned adjacent to related information in the main body area.
Default Header/Footer	Once the background logic is complete.	Setting the height of header and footer areas. Adding default content for headers/footers (which is used unless a Page Header/Footer is encountered).
At Start of Page	Once the default header/footer logic is complete.	Any processing to be performed only at the start of a page such as updating a variable. Note that this should contain logic only; so Active Content that produces physical content should not be included.
At End of Page	When a page in a document is about to end.	Any processing to be performed only at the end of a page such as incrementing counters. Note that this is primarily for logic – see note above – although you may include additional content on the current page provided it is included in a fixed anchor text box.

▶ To change header/footer content conditionally:



1. In the Page Setup view mark the insertion point in the logic map.

   You can use the *At Start of Page* and *At End of Pag*e groups to update the header/footer contents.

2. Create a condition object.

   You will typically want to use a Page Check object to set the header/footer according to the current page number within the document. You can also use a regular condition, case or loop object as required. See “Creating conditional logic” on page 132 for details.

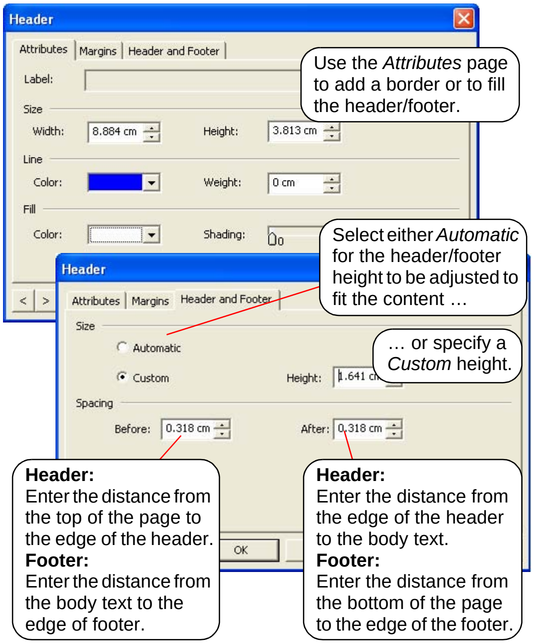
3. Create a new Header or Footer Content object.

   Mark the insertion point in the True or False branch of the condition and from the *Insert* menu select *Page Header* or *Page Footer* .

4. Add the new content.

   Click within the header/footer in the document editor view and create paragraphs and graphics as normal.

▶ To format header/footer areas: double-click Default Header/Footer or the appropriate Page Header/Footer object in the logic map. Use the *Header* or *Footer* dialog box to specify custom settings.



Label:

Size

Width: 8.884 cm

Height: 3.813 cm

Line

Color:

Weight: 0 cm

Fill

Color:

Shading: 0

Size

○ Automatic

● Custom

Height: 1.641 cm

Spacing

Before: 0.318 cm

After: 0.318 cm

Use the *Attributes* page to add a border or to fill the header/footer.

Select either *Automatic* for the header/footer height to be adjusted to fit the content …

… or specify a *Custom* height.

Header:
Enter the distance from the top of the page to the edge of the header.
Footer:
Enter the distance from the body text to the edge of footer.

Header:
Enter the distance from the edge of the header to the body text.
Footer:
Enter the distance from the bottom of the page to the edge of the footer.

Text that is inside a header or footer can be rotated 90° clockwise. You can rotate more than once until the text is in the required plane.

▶ To rotate text in a header/footer: select the text to be rotated in the header/footer area and on the *Format* menu select *Rotate Text* – or use the *Rotate* button in the Text Formatting toolbar.

▶ To change the header/footer size conditionally:

1. Mark the insertion point in the logic map.

   Header/footer size can only be changed within the Default Header or Default Footer Groups.

2. Create a condition object. See “Creating conditional logic” on page 132 for details.

3. Specify the new size.

   Mark the insertion point in the relevant branch of the condition and from the *Insert* menu select *Presentation Objects/Headers and Footers/Header Size* or *Footer Size*. Use the dialog box to enter the new size and adjust the spacing.

▶ To turn headers/footers off: double-click Page Setup in the Page Setup view and in the *Page Setup* dialog box, select *Turn Headers off /Turn Footers off*.

▶ To turn headers/footers on: double-click Page Setup in the Page Setup view and in the *Page Setup* dialog box, clear *Turn Headers off /Turn Footers off*.

Working with reserved areas

A reserved area defines a page space that has been set aside for dynamic update. For example, you may want to use a reserved area to create a personalized information box adjacent to a customer's transaction details. You could selectively add content to the box in response to particular types of transaction that are processed.

A reserved area is always defined as part of the Page Setup background layer but influences the presentation of objects within the body area where these overlap. Normally you will want your body content to flow around the reserved area although all the regular text flow options are available.

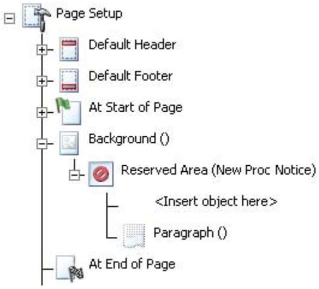
By default, reserved areas apply to all pages although you can use conditions within the Background to selectively include them on particular pages if required.

Initial content for a reserved area can be specified on the background if required. The content is updated by using a **Reserved Area Entry** object within your main document logic. When updating, you can choose to append to any existing material or clear the reserved area before adding the new content.

A reserved area always has the same position and size on every page on which it appears but you can create any number of reserved areas as required. A reserved area can contain any type of presentation object except for another reserved area.

In terms of positioning, color, line and fill attributes and text runaround properties, a reserved area behaves exactly the same as a text box or other shape. See "Working with graphics" on page 48 for details.

The initial content of a reserved area can be inserted in the Page Setup view either by entering it directly or by inserting a *Reserved Area* object in the *Background* object in the logic map. This content can be updated dynamically via the logic in the document view.



▶ To create a reserved area:

1. In the Page Setup view, mark the insertion pointer in the *Background* object in the logic map and from the *Insert* menu select *Tool Objects/ Reserved Area*.
2. Click in the page layout and drag the pointer where you want the box to be drawn.
3. Enter initial content (if any).

   You can create paragraph text and graphics as with a text box. Content will wrap and flow down the available space as with the main document body.

ⓘ IF THE RESERVED AREA IS NOT LARGE ENOUGH, IT WILL GROW DOWNWARDS AS YOU ADD NEW LINES IN THE **BACKGROUND**, BUT WILL NOT WRAP ONTO A NEW PAGE.

CONTENT ADDED VIA THE **DOCUMENT VIEW** WILL BE TRUNCATED IF IT DOES NOT FIT.

▶ To create a reserved area entry:

1. In the document view mark the insertion point.

   Click on an object in the logic map to indicate the point at which the update will be performed.

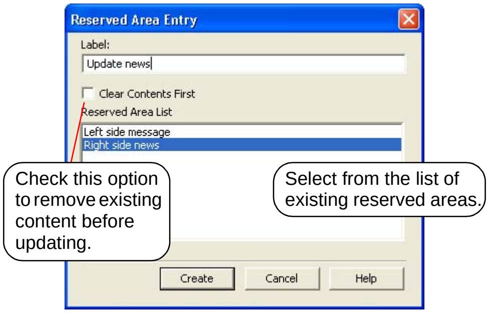
2. Select *Insert/Tool Objects/ Reserved Area Entry* from the menu.

3. Use the *Reserved Area Entry* dialog box to specify update settings.

4. Click *Create*.

5. Add new content.

   In the document editor the selected reserved area will become active. Create paragraphs and graphics as normal.



Label: Update news

☐ Clear Contents First

Reserved Area List

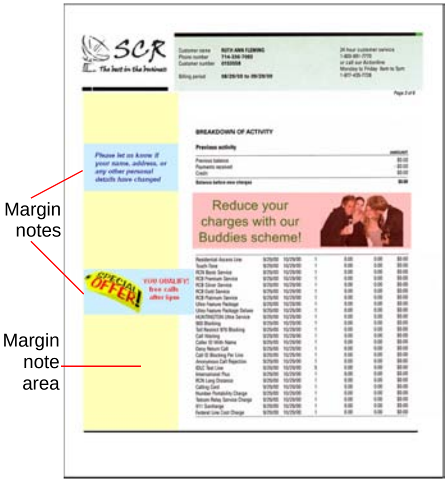
- Left side message
- Right side news

Check this option to remove existing content before updating.

Select from the list of existing reserved areas.

Working with margin notes

Margin notes are typically used to present related feature information adjacent to content in the main document flow but not interrupting it. For example, you may want to use a margin note to highlight particular information presented in a document that should be verified by the customer, or advertise a special offer that relates to a particular type of transaction.


Margin notes
Margin note area
Please let us know if your name, address, or any other personal details have changed
Reduce your charges with our Buddies scheme!
SPECIAL OFFER!

Margin notes are used in conjunction with a **margin note area,** which reserves space in the document design for controlling the horizontal placement of margin notes. Margin note areas are defined as part of the Page Setup background layer and have no presentation content of their own. The content is provided by the margin note, which is created within the main document logic.

Margin notes behave similarly to text boxes. You can place text, images and charts inside margin notes but you cannot insert message boxes, tables, reserved area entries, or Active Content.

You can place a margin note inside Active Content without a margin note area being present. The margin note will be included in the logic map but not shown in the document until an associated margin note area is created.

Margin notes are anchored to a paragraph in the main flow, or in a table cell. If the paragraph changes position the margin note will move accordingly. Similarly, if one overflows a page then they both move to the next page. The same behavior exists within a table cell.

ⓘ It is not recommended that margin notes are used with columns as the same margin note area is used for all columns on a page. Margin notes from the second and subsequent columns will be placed over those from previous columns.

If a margin note is vertically offset from its anchor to such an extent that it would be positioned off the page, or cause the overprinting of headers, the margin note will be positioned at the top of the flow area on a new page. Clicking on the margin note will display a screen indicator, which shows the relationship between a margin note and its anchor.

▶ **To create a margin note area:**

1. In *Page setup* view mark the insertion point in the *Background* object in the logic map.
2. From the *Insert* menu select *Tool Objects/ Margin Note Area*.

3. Click in the page layout and drag the pointer where you want the box to be drawn. The width of the area is set by the drawing action. The height is fixed to the extent of the page, excluding any header and footer.

▶ **To specify margin note area settings:** select the area and on the *Format* menu click *Shape*
– or –
right-click the *Margin Note Area* object in the logic map and select *Properties* from the shortcut menu.

Color, line and fill attributes, and text runaround properties are handled in a similar way to any other shape object. See "Working with graphics" on page 48.

ⓘ Be aware that use of a margin note area border will affect the positioning of margin notes in relation to the anchored paragraph.

If using multiple margin note areas in a document it is recommended that you use the *Attributes* tab to assign a unique label.

▶ **To create a margin note:**

1. In document view click within the paragraph that you want the margin note to be anchored to.

ⓘ You cannot create a margin note in a document unless a margin note area is present.

2. From the *Insert/ Presentation Objects* menu select *Margin note*
   – or –
   use the *Margin Note* button in the Objects toolbar.
3. Create the margin note box.

   Click in the page and drag out the shape to the required size. The margin note will automatically be positioned in the default margin note area, adjacent to its anchor object and will be resized to fit the width of the margin note area.
4. Add content as required.

   You can create paragraph text and graphics as with a text box.

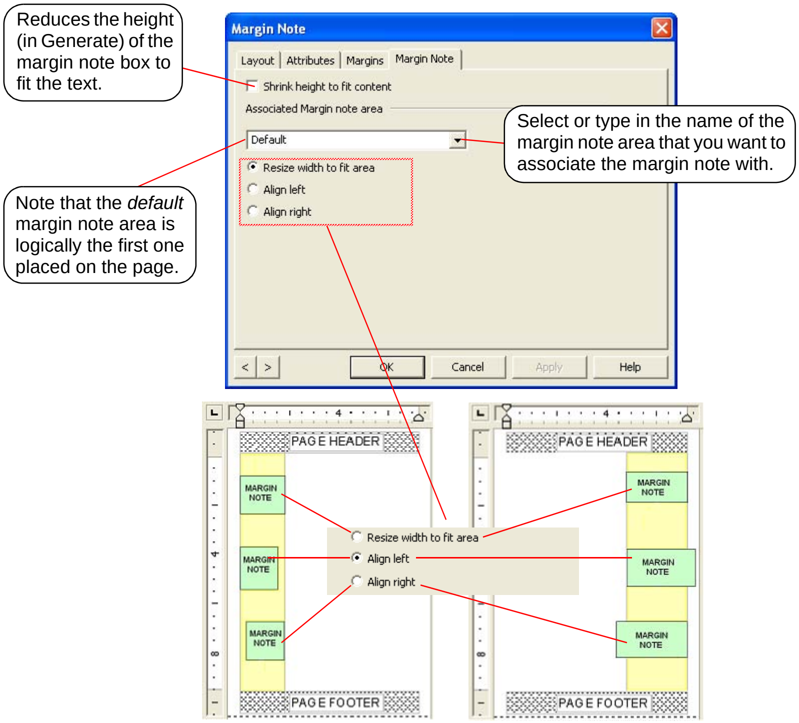
▶ **To specify margin note settings:**

1. Select the area and from the *Format* menu click *Shape*
   – or –
   right-click the *Margin Note* logic entry and select *Properties* from the shortcut menu.
2. The *Margin Note* dialog box is displayed. You can adjust margin note attributes as with any other shape object. See "Working with graphics" on page 48 for details.

   Note that you cannot change the width of the margin note area if the *Resize width to fit area* option is selected.

3. Use the *Margin Note* tab to set optimal height and width, adjust horizontal alignment, or to change margin note area association.

   If associating a margin note placed within Active Content that is wider than the margin note area, the content will **not** be reformatted. You will have to adjust the margin note manually.



Reduces the height (in Generate) of the margin note box to fit the text.

Note that the *default* margin note area is logically the first one placed on the page.

Select or type in the name of the margin note area that you want to associate the margin note with.

Margin Note

☐ Shrink height to fit content

Associated Margin note area: Default

● Resize width to fit area

○ Align left

○ Align right

Sections and page control

When new pages are generated they will, by default, use the page handling attributes specified for the document in the current Page Setup object and the initial Section in the Document view. These attributes include:

- Page orientation
- Page and sheet throws
- Header and footer use
- Column layout

By using additional sections you can override these defaults for particular parts of your document logic. For example, you may have a document design where the final page is an application form requiring landscape orientation, a different configuration of headers and footers, and which always needs to be printed on a new physical sheet on the printer.

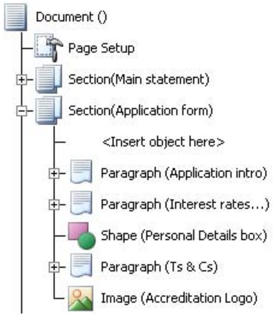
For such requirements you must include a section within your document logic and include presentation objects that require the special page attributes within the section object. Note that a document is automatically created with a single section and all presentation objects will be added to this group in the absence of other sections.

▶ **To create a new section:**

1. Mark the insertion point in the document logic.

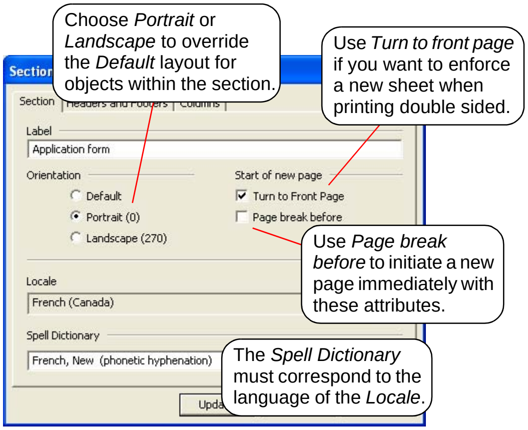
   The page handling attributes of the section will apply as soon as a new page is generated by the objects within it. You can specifically request a new page as part of the section itself if required.

2. On the *Insert* menu click *Section*.



Document ()
Page Setup
Section(Main statement)
Section(Application form)
<Insert object here>
Paragraph (Application intro)
Paragraph (Interest rates...)
Shape (Personal Details box)
Paragraph (Ts & Cs)
Image (Accreditation Logo)

3. In the *Section* dialog box specify the page handling attributes you require.



Label: Application form

Orientation: ○ Default ● Portrait (0) ○ Landscape (270)

Start of new page: ☑ Turn to Front Page □ Page break before

Locale: French (Canada)

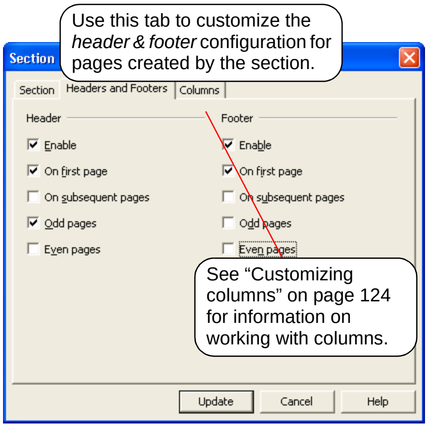
Spell Dictionary: French, New (phonetic hyphenation)

Choose *Portrait* or *Landscape* to override the *Default* layout for objects within the section.

Use *Turn to front page* if you want to enforce a new sheet when printing double sided.

Use *Page break before* to initiate a new page immediately with these attributes.

The *Spell Dictionary* must correspond to the language of the *Locale*.



Header: ☑ Enable ☑ On first page □ On subsequent pages ☑ Odd pages □ Even pages

Footer: ☑ Enable ☑ On first page □ On subsequent pages □ Odd pages □ Even pages

Use this tab to customize the *header & footer* configuration for pages created by the section.

See “Customizing columns” on page 124 for information on working with columns.

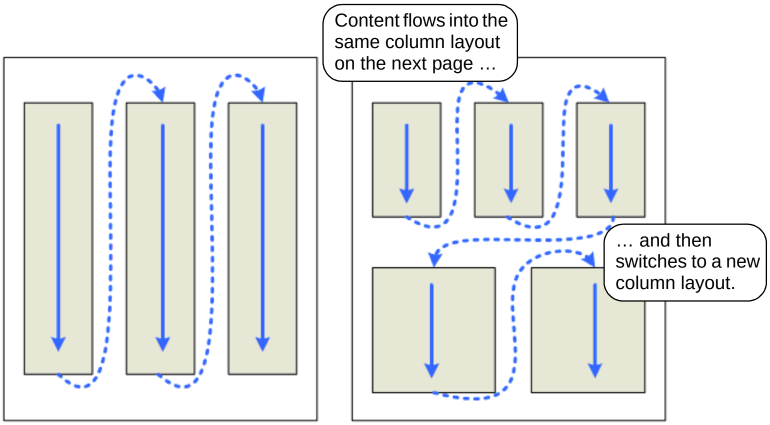
4. Add the objects to which the page handling attributes are to apply within the section.

> You can also specify a new default locale that will be active while the section is processed. See “Localization” on page 183 for details.

Customizing columns

SEE "COLUMNS" ON PAGE 64 FOR INFORMATION ON GENERATING SIMPLE COLUMNS.

The body area that is used for positioning main document content can be split into multiple columns, up to a maximum of eight in a single layout. Content flows top-to-bottom within a column and when an overflow occurs will continue in the next column in sequence in the layout. The sequence can be either left-to-right or right-to-left. Content will automatically flow onto new pages as required, continuing to use the same column layout. You can switch to a different layout at any time.


Content flows into the same column layout on the next page …
… and then switches to a new column layout.

Headers and **footers** can be applied to columns. These can be used in addition to, or instead of, page level headers and footers – see "Headers and footers" on page 113. They can also span across all columns. You can specify different header or footer content for regular use, for where content flows into a new column, and for where content flows into a new page.

Column layouts

A **column layout** is used to define the appearance of columns on a page. You can specify any number of layouts for a publication, switching between them as required. Most column layouts will be designed automatically to occupy the available width and height of the body area but you can limit the dimensions if required. A **canvas** is used to define the overall body size within which the column layout will be designed. This enables you to specify layouts for any page size or orientation, irrespective of the current document's page sizes.

NOTE THAT WIDOW/ORPHAN, KEEP WITH NEXT AND KEEP TOGETHER PARAGRAPH SETTINGS ARE IGNORED WHEN SWITCHING COLUMN LAYOUTS.

Column layouts can be defined as part of a document section or whenever you switch a column layout in the document logic.
A column layout can be reused anywhere in the current document and any other document designs in the publication.

If no column layout is defined then the **default layout** is used. This consists of a single column with no headers or footers that fits in the available body space. The default layout cannot be edited or deleted.

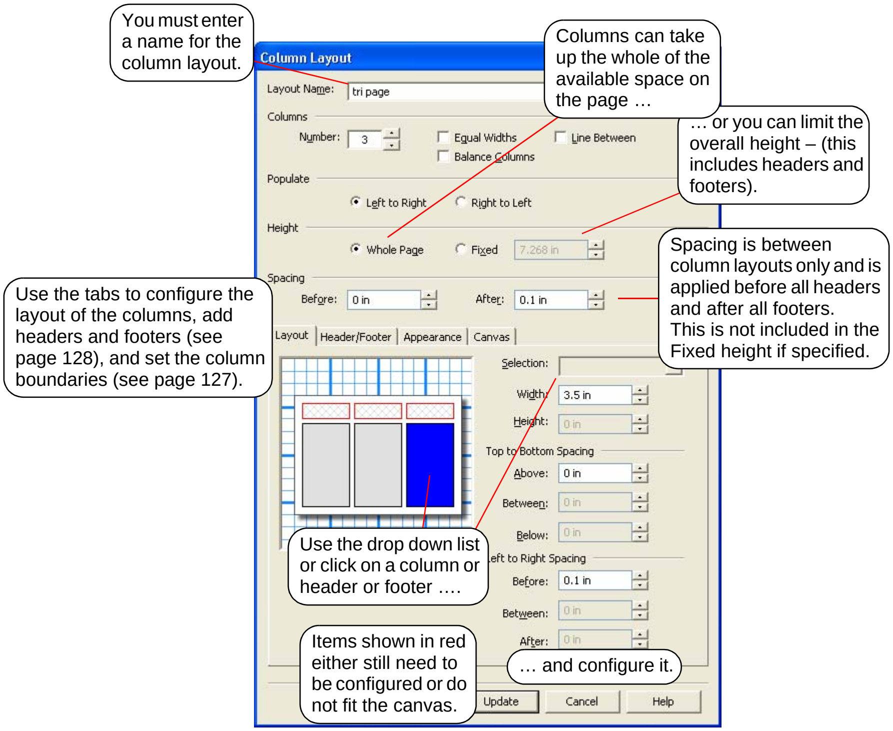
▶ To create or edit a column layout:

1. From the *Format* menu select *Column layout*
   – or –
   invoke the *Section* dialog box, click the *Columns* tab.
   – or –
   invoke the *Switch layout* dialog box (see “To switch column layouts’ on page 126).

> The *Section* dialog box is invoked either by double-clicking on *Section* in the logic map or by creating a new section using *Insert / Section*. See “Sections and page control” on page 169 for more information.

2. Select *New* or *Edit* and use the *Column layout* dialog box to configure the column layouts as required.

> Note that any changes made to the column layout will be applied to all instances where the layout is used.



Column Layout

Layout Name: tri page

Columns

Number: 3 ☐ Equal Widths ☐ Line Between

☐ Balance Columns

Populate

● Left to Right ○ Right to Left

Height

● Whole Page ○ Fixed 7.268 in

Spacing

Before: 0 in After: 0.1 in

Layout | Header/Footer | Appearance | Canvas

Selection:

Width: 3.5 in

Height: 0 in

Top to Bottom Spacing

Above: 0 in

Between: 0 in

Below: 0 in

Left to Right Spacing

Before: 0.1 in

Between: 0 in

After: 0 in

Update Cancel Help

You must enter a name for the column layout.

Columns can take up the whole of the available space on the page …

… or you can limit the overall height – (this includes headers and footers).

Spacing is between column layouts only and is applied before all headers and after all footers. This is not included in the Fixed height if specified.

Use the tabs to configure the layout of the columns, add headers and footers (see page 128), and set the column boundaries (see page 127).

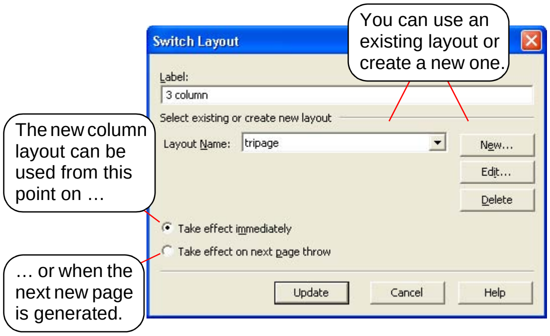
Use the drop down list or click on a column or header or footer ….

… and configure it.

Items shown in red either still need to be configured or do not fit the canvas.

▶ To switch column layouts:

1. In the document editor mark the insertion point in the logic map
   – or –
   click in the paragraph after which you want to specify the switch.
2. From the *Insert* menu select *Control objects/Columns/Switch column layout*. Select the required layout and choose when the change is to take effect.



Switch Layout

Label: 3 column

Select existing or create new layout

Layout Name: tripage

○ New... ○ Edit... ○ Delete

● Take effect immediately

○ Take effect on next page throw

Update | Cancel | Help

You can use an existing layout or create a new one.

The new column layout can be used from this point on …

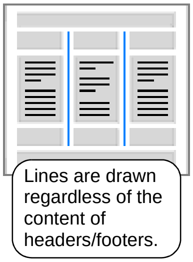
… or when the next new page is generated.

▶ To delete a column layout: invoke the *Section* dialog box, and click the *Columns* tab. Select a column layout and click *Delete*.

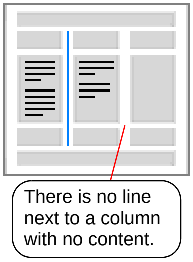
If you delete a column layout that is in use elsewhere in the publication then the default column layout will be used instead.

Line between columns

A line can be drawn between columns, giving them a visual separation. The line is central between columns and is drawn from the top of the column to the bottom, including any column headers and footers (but not spanning headers/footers). Lines will only be drawn between two columns with content, and will not be drawn next to an empty column. You can customize the color and weight of the line.



Lines are drawn regardless of the content of headers/footers.

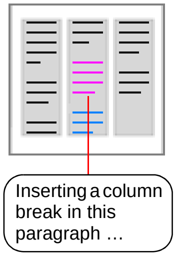


There is no line next to a column with no content.

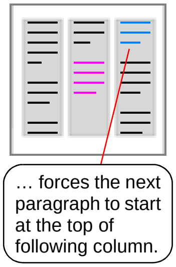
▶ To insert a line between columns: in the *Column layout* dialog box select the *Line between* option. Then select the *Appearance* tab and specify the color and weight as required.

Controlling column content

A column break can be used to force subsequent content of the column to start at the top of a new column, generating a new page if necessary. Note that a column break will take precedence over balancing.



Inserting a column break in this paragraph …

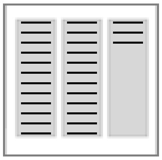


… forces the next paragraph to start at the top of following column.

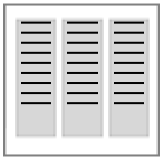
▶ To insert a column break:

1. In the document editor mark the insertion point in the logic map
   – or –
   position the cursor in the paragraph after which you want the column to break.
2. From the *Insert* menu select *Control objects/Columns/Column break*.

The content of columns on the final page of the layout can be balanced so that they are of similar height.



Unbalanced columns



Balanced columns

Note that the quality of balancing may be affected when used in conjunction with:

– tables

– widow and orphan settings

– anchored objects that are placed above the paragraph, or that overflow outside the column

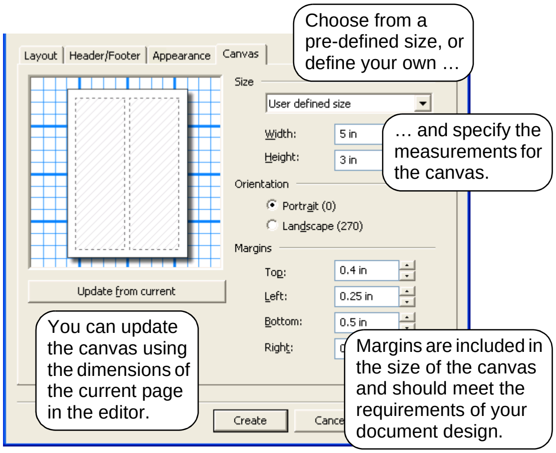
– shapes that are large in relation to the content to be balanced

▶ **To balance columns:** in the *Column layout* dialog box select the *Balance columns* option. Note that the use of balancing columns can cause reduction in performance time.

The column canvas

The column canvas allows you to specify the overall size of a column layout and is used for calculating the dimensions of the columns. You can use the size of the current page or specify a different size or orientation.

▶ **To specify a column canvas:** in the *Column layout* dialog box select the *Canvas* tab and set the options as required.



Size: User defined size

Width: 5 in

Height: 3 in

Orientation

● Portrait (0)

○ Landscape (270)

Margins

Top: 0.4 in

Left: 0.25 in

Bottom: 0.5 in

Right:

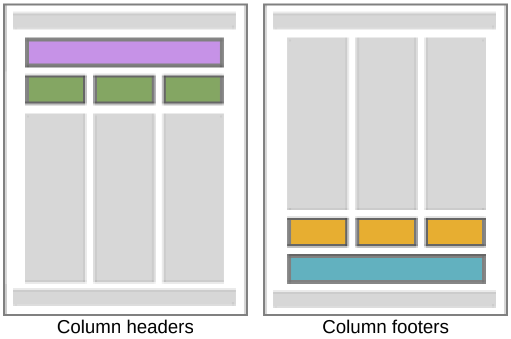
Choose from a pre-defined size, or define your own …

… and specify the measurements for the canvas.

You can update the canvas using the dimensions of the current page in the editor.

Margins are included in the size of the canvas and should meet the requirements of your document design.

Column headers and footers

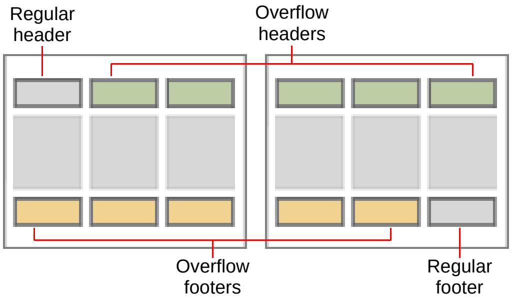


The headers/footers used in columns can all contain the same content or can use different content as required. This can be changed at any time in the processing, or when the columns overflow either into the next column or into the next page. This would allow, for example, 'Continued' to be placed in all headers except the first one. If required, headers and footers can also span across all columns in the layout, allowing you to use the entire width of the column layout for the content.

Column headers and footers are entirely separate from page headers and footers which can also be used on the same page.

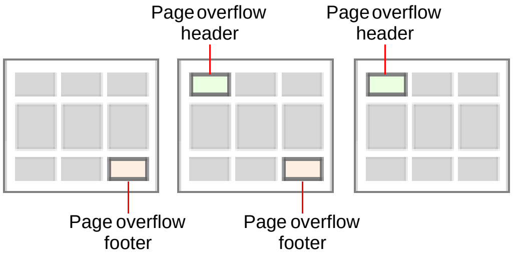
Overflow headers and footers

By selecting **overflow header** the regular column header will be used in the first column and the overflow header in all subsequent columns. By selecting **overflow footer** the regular column footer will be used in the last column and the overflow footer in all previous columns.

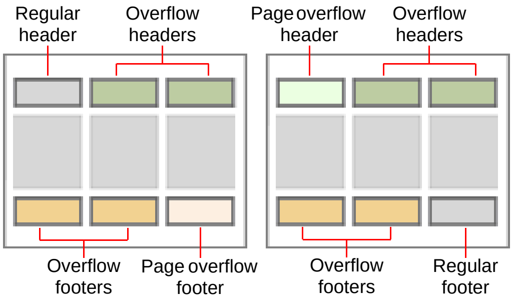


Page overflow headers and footers

The **page overflow header** is used in the first column of all pages except the first page and the **page overflow footer** is used in the last column of all pages except the last page. Other headers and footers are as required by the other options.

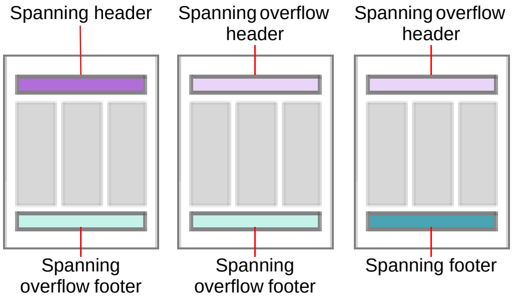


Note that if both column and page overflow options are specified, then the page overflow will take precedence over the column overflow.



Spanning overflow headers and footers

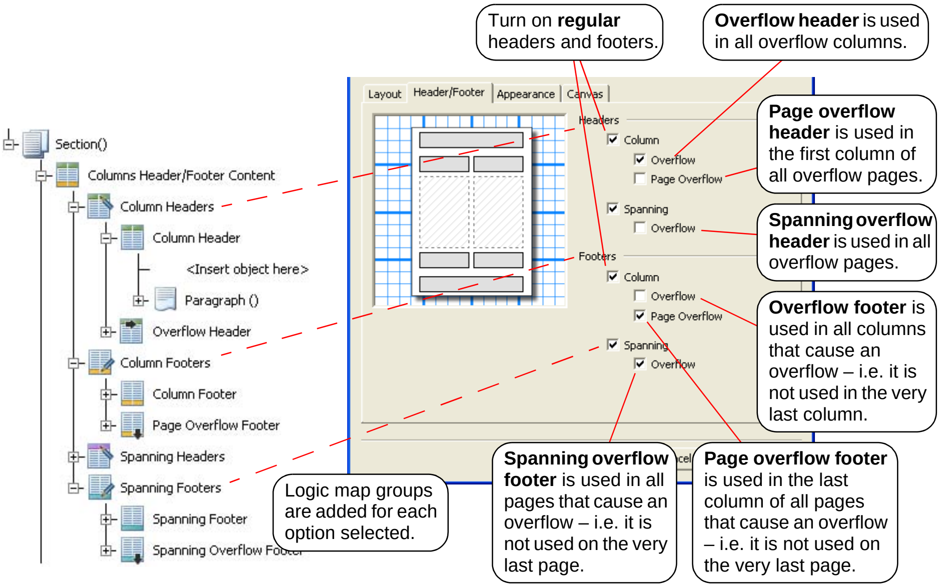
When **spanning overflow header** is selected the regular spanning header will be used on the first page and the spanning overflow header on all subsequent pages. When **spanning overflow footer** is selected the regular spanning footer will be used in the last page and the spanning overflow footer in all previous pages.



▶ To configure headers and footers:

1. In the *Column layout* dialog box select the *Header/Footer* tab and choose the required options.
2. In the *Layout* tab select the header or footer and specify its size and spacing.
3. Add content using the logic map or in the relevant header or footer in the document view.

   In the logic map a group for regular content will always be created when headers/footers are enabled. Additional logic map groups will be created for each overflow option selected. Each group should be separately populated with presentation objects – content is not shared between the groups.



Headers
☑ Column
☑ Overflow
☐ Page Overflow
☑ Spanning
☐ Overflow
Footers
☑ Column
☐ Overflow
☑ Page Overflow
☑ Spanning
☑ Overflow

Turn on **regular** headers and footers.

Overflow header is used in all overflow columns.

Page overflow header is used in the first column of all overflow pages.

Spanning overflow header is used in all overflow pages.

Overflow footer is used in all columns that cause an overflow – i.e. it is not used in the very last column.

Logic map groups are added for each option selected.

Spanning overflow footer is used in all pages that cause an overflow – i.e. it is not used on the very last page.

Page overflow footer is used in the last column of all pages that cause an overflow – i.e. it is not used on the very last page.

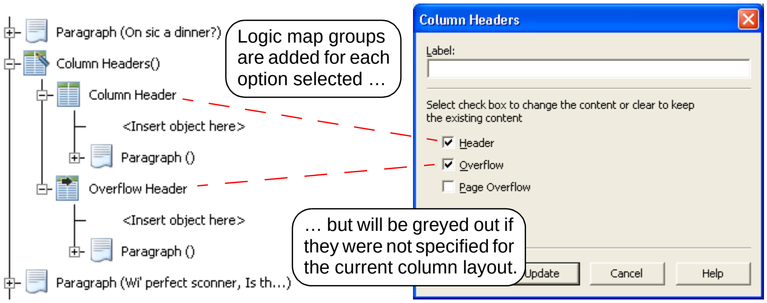
The content of headers and footers can be changed dynamically within the logic. The change will take effect the next time the header or footer is used and will apply to subsequent usage.

▶ To change the content of a column header or footer:

1. In the document editor mark the insertion point in the logic map
   – or –
   position the cursor in the paragraph after which you want the new header or footer to be used.

2. From the *Insert/Control objects/Columns* menu select either *Column headers, Column footers, Spanning headers* or *Spanning footers* and select which types of header or footer you want to change.

3. In the logic map a group for each selected type will be created. These should be separately populated with presentation objects – content is not shared between groups.



Label:

Select check box to change the content or clear to keep the existing content

☑ Header
☑ Overflow
☐ Page Overflow

Logic map groups are added for each option selected …

… but will be greyed out if they were not specified for the current column layout.

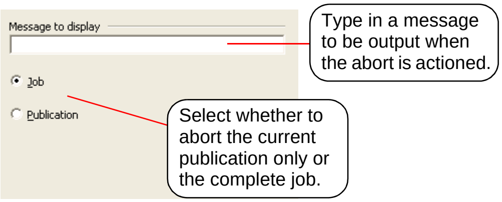
Using run-time actions

There are some general actions that affect the processing of a job at production time. They are specified using the **Action** object and can be inserted at most places in the logic.

▶ **To create an action:** mark the insertion point as required within your document logic and select *Insert/Control Objects/Action* from the menu. Select the required type in the *Action* dialog box:

Abort

Abort the current publication or the production job immediately, and output a message to the standard output device on your host machine.



Message to display

● Job

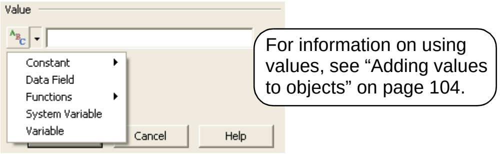
○ Publication

Type in a message to be output when the abort is actioned.

Select whether to abort the current publication only or the complete job.

Print message

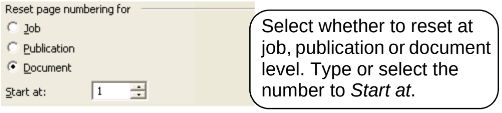
Output a message to the standard output device on your host machine. Use the value button to select a value be used for the message.



For information on using values, see “Adding values to objects” on page 104.

Reset page number

Reset page numbering at job, publication or document level and specify the page number you wish to start at.



Reset page numbering for

○ Job

○ Publication

● Document

Start at: 1

Select whether to reset at job, publication or document level. Type or select the number to *Start at*.

Note that this action causes inconsistent page counts to be generated for Page X of Y numbering and should be avoided if this system variable is present in your document design.

Creating conditional logic

DOC1 Designer has the following conditional logic features:

- *Condition* queries an expression and executes either a *True* or *False* logic map path depending on the result.
- *Page check* is a specialized version of a condition that allows you to query the current page number within a document or section to see if it matches those specified.
- *Loop* queries an expression and continuously executes a logic map path until it is no longer *True* or until a *Break Loop* is encountered. If required, *Continue Loop* can be used to restart the conditional logic without processing the remaining objects.
- *Case* selects one of a range of conditional paths when a comparison value is matched.

You may nest any of these objects within one another to form more complex logic as required.

YOU CAN ALSO USE THE *SHOW WHEN* ATTRIBUTE OF INDIVIDUAL PARAGRAPHS TO INCLUDE THEM CONDITIONALLY WITHOUT THE NEED TO USE ONE OF THE ABOVE OBJECTS. REFER TO "CONDITIONAL PARAGRAPHS" ON PAGE 44 FOR DETAILS.

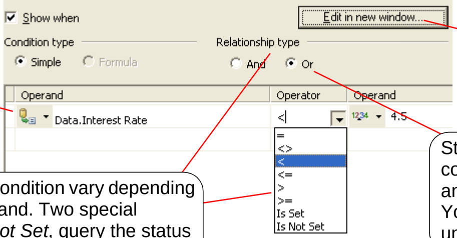
Conditional expressions

The expressions used in Condition and Loop objects can be specified either using a *Simple* method with arguments specified in a structured format or by coding the arguments using a *Formula* expression. Values can be specified as constants or be supplied by data fields, variables or calls to system or Active Content functions.

The values being queried must be of the same type but in many circumstances you can temporarily reformat the data to ensure consistency. See "Formatting values" on page 106 for details.

To code a simple condition click the *Simple* option and enter a comparison argument using the fields provided in the object dialog. Use the operand controls to specify input values for the comparison.

You may add further arguments as required. If the AND operator is specified all arguments must be true for the *True* logic to be executed. If the OR operator is specified the *True* logic will be executed if any of the arguments are true.



☑ Show when

Edit in new window...

Condition type: ● Simple ○ Formula

Relationship type: ○ And ● Or

Operand	Operator	Operand
Data.Interest Rate	<	1234 4.5

=
<>
<
<=
>
>=
Is Set
Is Not Set

Click to select a value type and name. See “Adding values to objects” on page 104 for details.

This allows you to resize the window making it easier to view long or complex formula conditions.

The operators available in a condition vary depending on the value type of the operand. Two special comparisons, *Is Set* and *Is Not Set*, query the status of a data field (to see if it is available in the current data set) or a variable (to see if it is currently in scope).

Statements in simple conditions can have either an *And* or an *Or* relationship. You cannot use both types unless you code a Formula.

To code a formula click the *Formula* option and enter the appropriate syntax.

ⓘ FOR DETAILED INFORMATION SEE “APPENDIX B – FORMULA EXPRESSIONS” ON PAGE 414.

Condition

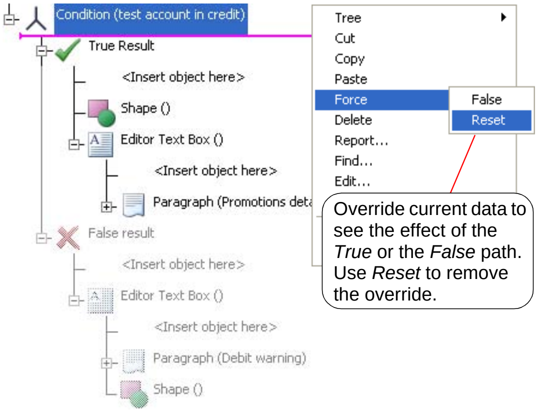
A condition object queries an expression and executes either a *True* or *False* logic map path depending on the result.

▶ To create a condition:

1. Mark the required insertion point within the logic map.
2. Select *Insert/Control Objects/Condition/Condition* from the menu.
3. In the *Condition* dialog select either *Simple* or *Formula*.
4. Create the required query. See “Conditional expressions” on page 132 for details.
5. Click *Create* and populate the logic tree.

In the logic map a condition object is automatically created with True and False paths. Add the required logic to each.

To help when designing a document, you can force the True or False path to be active. This enables you to see the effect of each path in the condition. You can reset all paths in the document by selecting *Insert/Control Objects/Condition/Reset All*.



Override current data to see the effect of the *True* or the *False* path. Use *Reset* to remove the override.

▶ **To delete a simple condition argument:** in the *Condition* dialog box select the argument you want to remove and press *Delete*.

Page check

Page check is a specialized version of a condition that allows you to query the current page number within a document or section to see if it matches those specified. You can perform a simple check to see if the current page number is odd or even or you can reference one or more specific pages either individually or within ranges.

As with a regular condition, a *True* or *False* logic map path is executed depending on the result of the check.

▶ **To use a page check:**

1. Mark the required insertion point within the logic map.
2. Select *Insert/Control Objects/Condition/Page Check* from the menu.
3. In the *Page Check* dialog select either *Job*, *Publication* or *Document* to indicate the scope of the check.

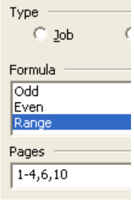
   In each of these, the page check assumes a starting point of page 1.
4. Specify the page formula to be matched.

   Selecting *Odd* will mean the True path is executed for page 1, 3, 5, etc. Similarly, pages 2, 4, 6, etc. will execute the True path when *Even* is selected.

   If *Range* is selected, indicate the page numbers that will cause the True path to be executed in the entry field, for example:

   ```
   1,7,11
   1-4,8
   1-10,20+
   ```
5. Click *Create* and populate the logic tree.

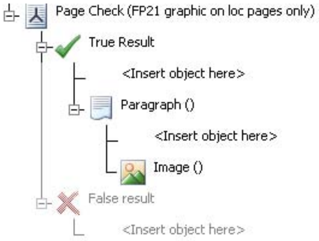
In the logic map a page check object is automatically created with True and False paths. Add the required logic to each.



Type
○ Job

Formula
Odd
Even
Range

Pages
1-4,6,10



Page Check (FP21 graphic on loc pages only)
True Result
<Insert object here>
Paragraph ()
<Insert object here>
Image ()
False result
<Insert object here>

Loop

A loop object queries an expression and continuously executes a logic map path until it is no longer *True* or until a *Break Loop* is encountered. If required, *Continue Loop* can be used to restart the conditional logic without processing the remaining objects. Loops can be nested within each other.

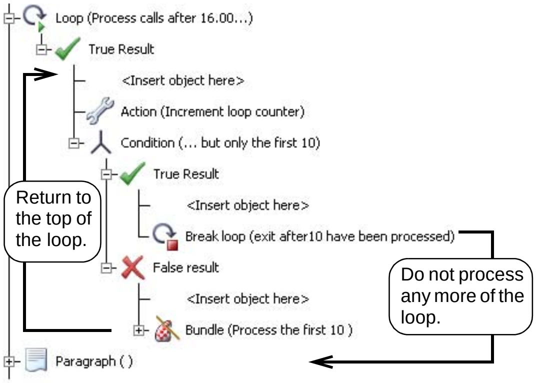
If the conditional expression does not evaluate to true when the loop object is first encountered the True path will not be executed.

▶ To create a loop:

1. Mark the required insertion point within the logic map.
2. Select *Insert/Control Objects/Loop/Conditional Loop* from the menu.
3. In the *Loop* dialog box select either the *Simple* or *Formula* condition type and set the *Maximum Loop Iterations*.
4. Create the required query. See “Conditional expressions” on page 132 for details.
5. Click *Create* and populate the logic tree.

In the logic map a loop object is automatically created with a True path. Add the objects that you want to be processed while the conditional expression is true.

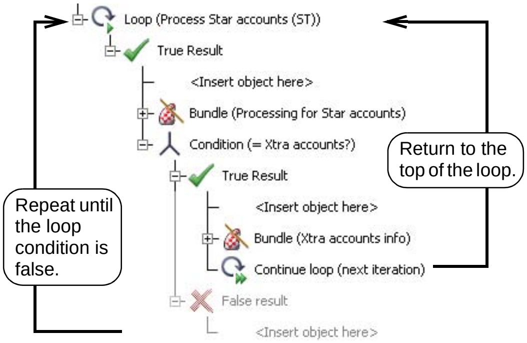
▶ To break out of a loop: mark the insertion point within the True path and select *Insert/Control Objects/Loop/Break* from the menu.



Loop (Process calls after 16.00...)
True Result
<Insert object here>
Action (Increment loop counter)
Condition (... but only the first 10)
True Result
<Insert object here>
Break loop (exit after10 have been processed)
False result
<Insert object here>
Bundle (Process the first 10)
Paragraph ( )

Return to the top of the loop.
Do not process any more of the loop.

▶ **To include specific processing within a loop:** you will typically want to use another conditional object within the True path to evaluate a specific status and include any specific processing. On the appropriate path within the embedded object mark the insertion point and select *Insert/Control Objects/Loop/Continue* from the menu.



Loop (Process Star accounts (ST))
True Result
<Insert object here>
Bundle (Processing for Star accounts)
Condition (= Xtra accounts?)
True Result
<Insert object here>
Bundle (Xtra accounts info)
Continue loop (next iteration)
False result
<Insert object here>

Repeat until the loop condition is false.

Return to the top of the loop.

Case

A case object selects one of a range of conditional paths when a comparison value is matched. The value is defined within the main case object and is compared with the values defined for the *Case Test* objects within the case group. The logic associated with the first case test that matches the comparison value is executed. If no value is matched an *Otherwise Result* logic path is executed.

Values can be specified as constants or be supplied by data fields or any other value object. All values must be of the same type but in many circumstances you can temporarily reformat the data to ensure consistency. See “Formatting values” on page 106 for details.

▶ **To create a case group:**

1. Mark the required insertion point within the logic map.
2. Create the case object and supply the comparison value.

   Select *Insert/Control Objects/Condition/Case* from the menu.

   In the *Case* dialog box give the object a reference *Label* and provide the value used for the comparison, see “Adding values to objects” on page 104 for details.

3. Create case tests.

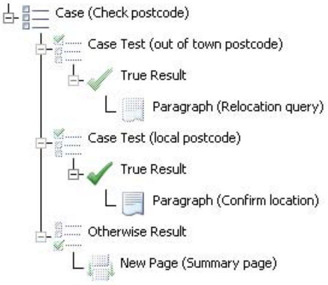
   Mark the insertion point on <Insert object here> and select *Insert/Control Objects/Condition/Case Test* from the menu. The expanded case test object will contain a place holder and a True Result group.

   Resolve the place holder with an object containing the value that will be compared against the main case value.

   Add the logic to be executed when the value is matched to the *True* path

   Repeat for each case test required.

4. If required, add logic to the *Otherwise* path that will be executed if no case test has been matched.



Case (Check postcode)
Case Test (out of town postcode)
True Result
Paragraph (Relocation query)
Case Test (local postcode)
True Result
Paragraph (Confirm location)
Otherwise Result
New Page (Summary page)

Data driven character styles

You can use style lists to change the presentation of text dynamically according to conditional logic.

Style lists are objects that you can place within your document logic and are themselves containers for one or more *styles*.

A style itself contains the actual text formatting instructions. It always specifies a font and point size and can also include font decoration settings such as color, bold, underline and so on. A style can be applied to most text elements including data fields, address blocks and charts.

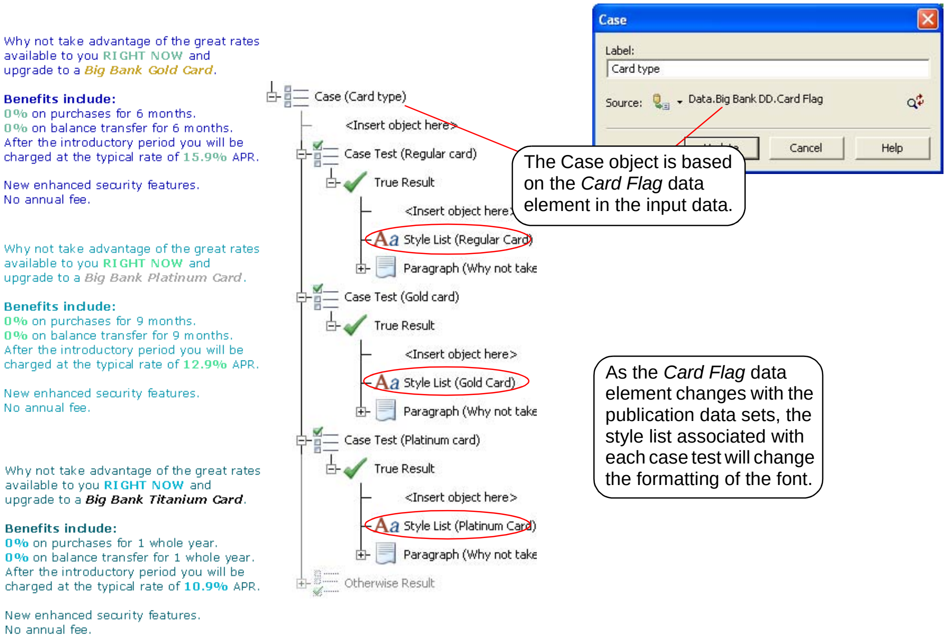
CHARACTER STYLES ARE NOT APPLIED CUMULATIVELY, I.E. YOU CANNOT APPLY TWO OR MORE STYLES TO THE SAME TEXT AND GET THE JOINT EFFECT OF BOTH. ONLY THE LAST STYLE APPLIED WILL BE USED. THEREFORE EACH STYLE SHOULD HAVE ALL THE SETTINGS REQUIRED TO PRESENT THE TEXT ACCORDING TO YOUR NEEDS.

At production time the active style list is the last such object to be encountered in the logic. Before a style list is encountered no styles are used and all text elements are presented using their regular settings. Once a style list is encountered any text that follows the object within the logic will adopt the settings of the styles it contains wherever a style name from the list has been applied. Subsequent style lists within the logic may amend some or all of the named styles. If a style name does not appear in a style list its settings remained unchanged from the previous time a style list included it.

Style lists are not stored in the repository as independent files. However, you can reuse the logic you set up to control styles by including it in an Active Content file and including this in any design where you need the same functionality.

You will typically want to use conditional objects to choose when to invoke particular style lists. Remember to make sure that text to be influenced by a style list appears after the list objects within the document logic.

In the following example a case object is used to create the necessary conditions for dynamically changing the character styles.



Why not take advantage of the great rates available to you RIGHT NOW and upgrade to a *Big Bank Gold Card*.

Benefits include:
0% on purchases for 6 months.
0% on balance transfer for 6 months.
After the introductory period you will be charged at the typical rate of 15.9% APR.

New enhanced security features.
No annual fee.

Why not take advantage of the great rates available to you RIGHT NOW and upgrade to a *Big Bank Platinum Card*.

Benefits include:
0% on purchases for 9 months.
0% on balance transfer for 9 months.
After the introductory period you will be charged at the typical rate of 12.9% APR.

New enhanced security features.
No annual fee.

Why not take advantage of the great rates available to you RIGHT NOW and upgrade to a ***Big Bank Titanium Card***.

Benefits include:
0% on purchases for 1 whole year.
0% on balance transfer for 1 whole year.
After the introductory period you will be charged at the typical rate of 10.9% APR.

New enhanced security features.
No annual fee.

Case

Label: Card type

Source: Data.Big Bank DD.Card Flag

The Case object is based on the *Card Flag* data element in the input data.

As the *Card Flag* data element changes with the publication data sets, the style list associated with each case test will change the formatting of the font.

▶ To set character styles dynamically:

1. In the document's logic map insert a Case object and create a case group.

   For detailed information on case groups, see "Case" on page 136.

> ⓘ NOTE THAT YOU CAN ALSO USE OTHER CONDITIONAL LOGIC, SUCH AS PAGE CHECKS, CONDITIONS AND LOOPS.

2. For each case test, mark the insertion point in the logic map and select *Insert/Control Objects/Set Style* to place a Style List object.
3. Double-click the initial *Style List* object in the logic map and the *Style List* dialog box is displayed.
4. Click *Add* to display the *Set Style* dialog box.

5. Enter a name (to a maximum of 32 characters) for the style created in the *General* tab and then select the *Font* tab to set the attributes for the character style.

   Repeat the step for each character style required in the document.

> THE INITIAL STYLE LIST OBJECT IS USED TO SET UP THE CHARACTER STYLES THAT WILL BE DYNAMICALLY CHANGED BY EACH CASE TEST.

6. For each subsequent *Style List* object repeat steps 5 to 7, however, in the *General* tab of the *Set Style* dialog box select one of the existing character styles from the list and then select the *Font* tab to set its attributes.

7. Apply the character styles to the necessary text.

   Highlight the appropriate text in the editor and select the character style from the list in the toolbar
   – or –
   highlight the text and right-click it to display the shortcut menu. Select *Font* to display the *Font* dialog box and click the *Style* option to select a character style from the list.

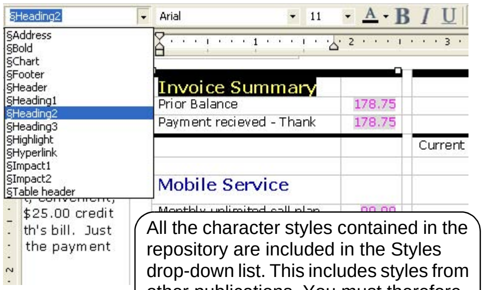
> ALL THE CHARACTER STYLES CONTAINED IN THE REPOSITORY ARE AVAILABLE FOR SELECTION. THIS INCLUDES STYLES FROM OTHER PUBLICATIONS, MEANING YOU MUST ENSURE YOU USE CHARACTER STYLES AVAILABLE TO YOUR DOCUMENT FOR THE FORMATTING TO TAKE EFFECT.

Using character styles

After creating style lists and character styles you can apply them to text, as well as edit and remove them from Style Lists.

▶ To apply a character style to text:

1. Highlight the appropriate text in the editor and select the character style from the list in the toolbar
   – or –
   highlight the text and right-click it to display the shortcut menu. Select *Font* to display the *Font* dialog box and click the *Style* option to select a character style from the list.



All the character styles contained in the repository are included in the Styles drop-down list. This includes styles from other publications. You must therefore ensure that you use character styles that are available to your document for the formatting to take effect.

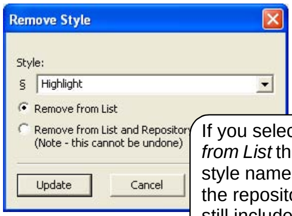
▶ To modify an existing style:

1. Double-click the Style List object in the logic map.
2. Select a character style and click *Edit* to display the *Set Style* dialog box.
3. Choose the *Font* tab and click *Select* to make the necessary changes to the font attributes and click *OK*.

4. Select a different character set from the list, if necessary, and click *Update*.

▶ **To remove a character style from the style list:**

1. Invoke the *Style List* dialog box and click *Remove* to display the *Remove Style* dialog box.
2. Select the character style to remove from the drop-down list.
3. Select either *Remove from List* or *Remove from List and Repository* and click *Update*.



Style: § Highlight

● Remove from List

○ Remove from List and Repository (Note - this cannot be undone)

If you select *Remove from List* the character style name remains in the repository and is still included in style lists for other objects.

▶ **To remove a character style from text:** highlight the text that you want to remove the character style from. Change the font attributes as required using the toolbar options.

Functions

The DOC1 Designer includes the following functions:

Address block: this function automatically formats address data based on the chosen country. The resulting address structure is output and positioned as a single object

Arithmetic: performs a calculation on two input values and outputs the result.

Barcodes: converts strings and numbers to the necessary sequence of lines required for the chosen barcode type.

Concatenation: this function enables you to join text values from two or more objects to form a single string.

Lookup table: enables you to replace predefined keywords with text strings at production time.

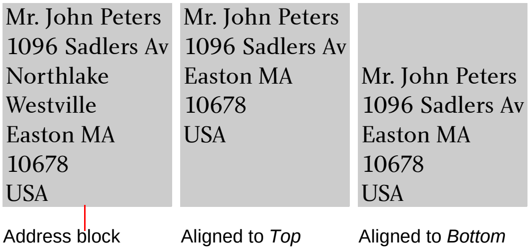
Substring: this function extracts one or more consecutive characters from within a text string.

System variables: can be used to access run-time statistics and information from the productions environment.

Address block

The address block function automatically formats name and address data into a style that is the standard for a named country. The resulting address structure is output and positioned as a single object.

The output lines of the address block are automatically aligned vertically and the number of lines actually placed is adjusted to compensate for empty input values. The output lines are placed within a notional outline that allows enough horizontal space to accommodate all possible address lines. You can choose to align the block either at the top or bottom of the outline space.



Address block outline space

Aligned to *Top*

Aligned to *Bottom*

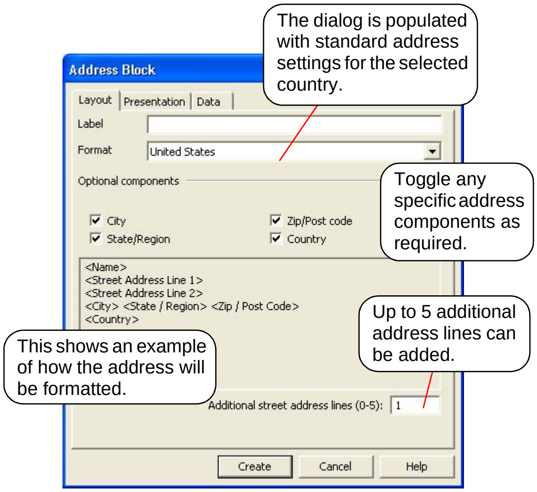
The font and line spacing used for the output lines is set for the entire object although you can override the settings of individual components as required. While you can change the font style, size and decoration for the text of an address block, you cannot change the justification or add bullets and numbering.

In terms of positioning, color, line and fill attributes and text runaround properties, an address block behaves exactly the same as a text box or other shape. See "Working with graphics" on page 48 for details.

▶ To insert an address block:

1. In the document editor mark the insertion point in the logic map
   – or –
   position the cursor in the paragraph to which the address block is to be anchored.
2. From the *Insert* menu click *Presentation Objects/Address Block*.
3. In the *Address Block/Layout* dialog box use the *Format* list to select the country for which the address is to be formatted.
4. On the *Presentation* page select the font, text alignment, direction and line spacing to be used for output lines unless overridden.

   Line spacing is either:
   Automatic – the line spacing value contained in the default font
   Absolute – a fixed gap between lines that you define. This spacing will be applied regardless of the font size.



Label
Format: United States
Optional components
☑ City ☑ Zip/Post code
☑ State/Region ☑ Country

<Name>
<Street Address Line 1>
<Street Address Line 2>
<City> <State / Region> <Zip / Post Code>
<Country>

Additional street address lines (0-5): 1

The dialog is populated with standard address settings for the selected country.

Toggle any specific address components as required.

Up to 5 additional address lines can be added.

This shows an example of how the address will be formatted.

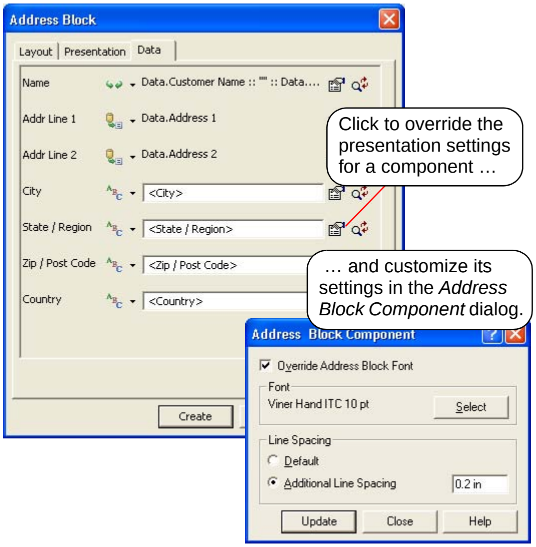
5. On the *Data* page resolve all the address component fields with values.

6. Select the *Create* button to place the object on the page.

   Once placed you can reposition it and alter its font and properties as required.

▶ **To change the font of a component in an address block:** highlight the component and select the required font, size, color etc.

▶ **To change the properties of an address block:** right-click on its border and from the shortcut menu select *Format Shape*. Use the dialog box to change the properties. See “Working with graphics” on page 48 for more information.



Name: Data.Customer Name :: "" :: Data....
Addr Line 1: Data.Address 1
Addr Line 2: Data.Address 2
City: <City>
State / Region: <State / Region>
Zip / Post Code: <Zip / Post Code>
Country: <Country>

☑ Override Address Block Font
Font: Viner Hand ITC 10 pt
Line Spacing
○ Default
● Additional Line Spacing: 0.2 in

Click to override the presentation settings for a component …

… and customize its settings in the *Address Block Component* dialog.

Arithmetic

An arithmetic function performs a calculation on two input values and outputs the result. Each value can be of any type, but must be cast to the same type as the result, see “Casting” on page 109 for details.

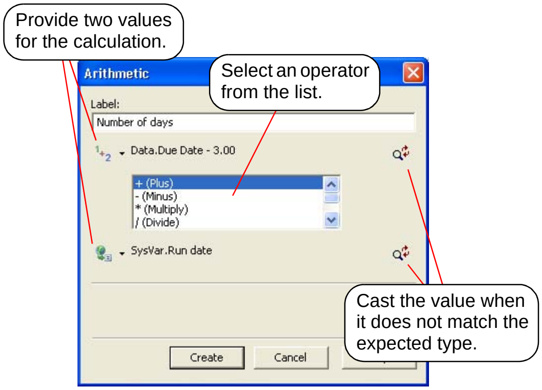
▶ **To create an arithmetic function:**

1. From the *Insert* menu select *Field/Arithmetic* and proceed to step 3
   – or –
   create an object to receive the output.

   You can update a number or date variable with the function output or use it as input to any other object that requires a number or date value. See “Working with values” on page 104 for details.

2. When specifying a value for the object select *Functions/Arithmetic* as the type.

3. In the *Arithmetic* dialog box specify the calculation.



Label: Number of days

Data.Due Date - 3.00

+ (Plus)
- (Minus)
* (Multiply)
/ (Divide)

SysVar.Run date

Provide two values for the calculation.

Select an operator from the list.

Cast the value when it does not match the expected type.

Arithmetic operators

Plus Valid for numbers, integers and dates, the result will be the addition of the values– i.e. A+B.

Minus Valid for numbers, integers and dates, the result will be the subtraction of the second value from the first value – i.e. A-B.

Multiply Valid for numbers and integers, the result will be the multiplication of the values – i.e. A*B.

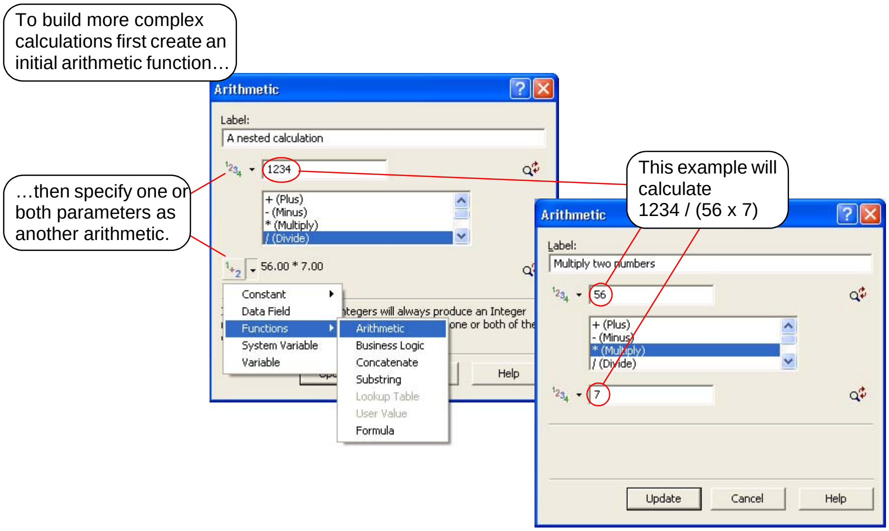
Divide Valid for numbers and integers, the result will be the division of the values – i.e. A/B. Note that if both values are integer, then the result will be integer, i.e. truncated; however, if at least one of the values is a number, then the result will also be a number.

Modulo Valid for integers only, the returned value will be the remainder that results from the modular division of the values – i.e. (A/B). For example 10/3 gives a remainder, or modulo, of 1.

Bitwise AND' Valid for integers only, 'AND' compares the two values bit by bit. If every bit in position ‘x’ of <u>both</u> values is 1, bit ‘x’ in the result is set to 1, otherwise 0.

Bitwise OR' As above except that if a bit in position 'x' of <u>either</u> of the values is 1, bit 'x' in the result is set to 1.

You can nest arithmetic functions to perform more complex equations if required. The evaluation sequence will follow the order in which the objects are nested, from top to bottom as in the following example.



To build more complex calculations first create an initial arithmetic function…

…then specify one or both parameters as another arithmetic.

This example will calculate 1234 / (56 x 7)

Arithmetic

Label: A nested calculation

1234

/ (Divide)

56.00 * 7.00

Arithmetic

Label: Multiply two numbers

56

* (Multiply)

7

Date calculations

Calculations can be performed on date values with the following restrictions:

- The operator used must be either addition or subtraction.
- The first value must contain a date value.
- The second value indicates the time to add or subtract from the date. This can be specified as:

an integer	– of the number of days
a string	– indicating the number and type of time units
a date	– for subtraction only.

Examples of time units:
2W = two weeks
5M = five months
3Y = three years.

Barcodes

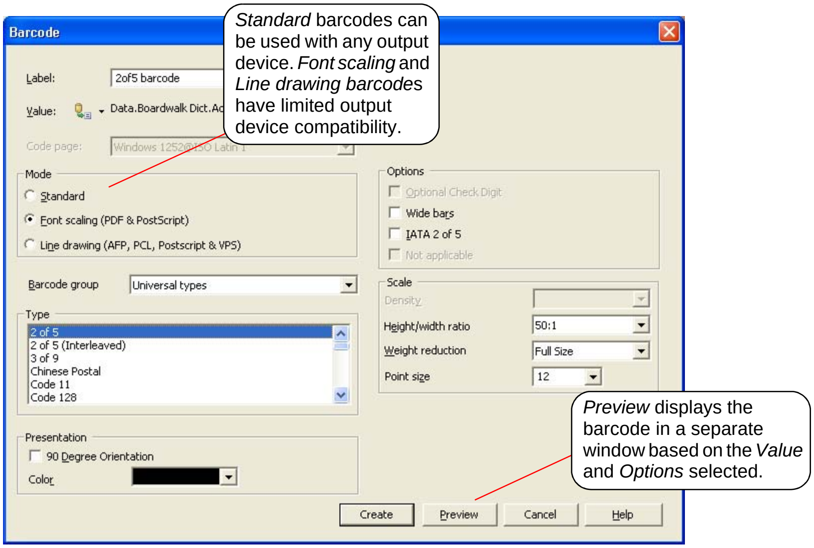
The barcode function converts strings and numbers to the appropriate sequence of lines that are required for the selected barcode. DOC1 includes support for *standard*, *font scaling* and l*ine drawing* barcode types, as the barcode fonts are automatically installed in the work center and available for all drivers and variants. See below for the list of barcodes and their settings.

Standard and *font scaling* barcodes generate textual output and make reference to special fonts within the output datastream produced by DOC1 Generate. As with all other text output the appropriate fonts will be automatically included in the HIP file that controls a DOC1 production job once it is published. By default these and all other fonts are embedded in the output datastream, but if you choose not to embed you will need to extract the resources from the HIP file and make them available to your output device.

> Refer to “Running a publishing task” on page 336 for details of font embedding options and to the DOC1 Programmer’s Guide for information on using DOC1RPU to extract resources from a HIP file

Line drawing barcodes are drawn without the use of fonts; native drawing methods are used instead, such as lines or rectangles.

In terms of positioning, color, line and fill attributes and text runaround properties, a barcode behaves exactly the same as a text box or other shape, see “Working with graphics” on page 48 for details.



Barcode

Label: 2of5 barcode

Value: Data.Boardwalk Dict.Ac

Code page: Windows 1252@ISO Latin 1

Mode
- ○ Standard
- ● Font scaling (PDF & PostScript)
- ○ Line drawing (AFP, PCL, Postscript & VPS)

Barcode group: Universal types

Type: 2 of 5 (selected); 2 of 5 (Interleaved); 3 of 9; Chinese Postal; Code 11; Code 128

Presentation
- □ 90 Degree Orientation
- Color: ■

Options
- □ Optional Check Digit
- □ Wide bars
- □ IATA 2 of 5
- □ Not applicable

Scale
- Density:
- Height/width ratio: 50:1
- Weight reduction: Full Size
- Point size: 12

Standard barcodes can be used with any output device. *Font scaling* and *Line drawing barcode*s have limited output device compatibility.

Preview displays the barcode in a separate window based on the *Value* and *Options* selected.

▶ To create a barcode:

1. Place the cursor in the document area where you want the barcode.
2. Select *Insert/Presentation Objects/Barcode* and the *Barcode* dialog box is displayed.

3. Select the *Standard*, *Font scaling* or *Line drawing* mode of operation.
4. Select the required barcode and resolve any barcode specific settings.

▶ **To change the properties of a barcode:** right-click on its border and from the shortcut menu select *Shape*. Use the *Shape* dialog box to change the properties. See “Working with graphics” on page 48 for more information.

Standard barcodes

Standard barcodes can be used with any output device. To be compatible with each output device there are a limited number of barcodes and associated options available when using standard mode. Apart from changing the density, to enable other scalability options you must use font scaling barcodes.

Standard barcode settings

Label	Enter an optional name for the barcode object that will appear in the logic map.
Value	Enter the value you want to use for the barcode. See “Adding values to objects” on page 104 for more information.
Mode	*Standard* mode barcodes can be used with any output device. For *Font scaling barcodes* see page 148 and for *Line drawing barcodes* see page 151.
Barcode group	Select *Universal types* or one of the 2D barcode groups to populate the *Type* list.
Type	If *Universal types* has been selected, this list is populated with standard barcodes, otherwise with the 2D barcodes available for the selected *barcode group*.
Scale	*Density* can be set to Low, Medium or High.
Presentation	These are cosmetic options that affect the presentation of the barcode when it is placed on the page. *90 degree orientation* rotates the barcode 90 degrees clockwise. Use the *Color* option to change the color of the barcode from the default black.
Specific barcode options	
PDF417 options	*Code page* – if you need to include non-Latin characters in the barcode this allows you to select an alternative code page. *Security* defines the amount of additional code words in the final barcode which can be used to reconstruct the barcode if the barcode has been damaged or misread. *Barcode aspect* is the height to width ratio of the number of code words per row. 0 will provide a tall, thin barcode and 100 will provide a short, wide barcode. *Start mode* indicates the type of data most likely to occur at the start of the data to be encoded. *Font aspect* controls the aspect of the font to be used to present the barcode. *Weight reduction* enables the font weight of the barcode to be changed.

(continued)

Standard barcode settings

Datamatrix options

Code page – if you need to include non-Latin characters in the barcode this allows you to select an alternative code page.
Mode determines the shape of the final barcode.
Security defines the amount of additional codewords in the final barcode which can be used to reconstruct the barcode if the barcode has been defaced or misread.
Weight reduction enables the font weight of the barcode to be changed.

QR Code options

Code page – if you need to include non-Latin characters in the barcode this allows you to select an alternative code page. This is only available when the mode is set to *binary*.
Mode determines which type of data will be used with the barcode.
Security defines the amount of additional code words in the final barcode which can be used to reconstruct the barcode if the barcode has been defaced or misread.
Weight reduction enables the font weight of the barcode to be changed.

Aztec

Mode determines the characteristics of the final barcode. Choose from:
Normal – allows you to set *Security* to any percentage in the range 1 – 99%.
Compact – use *Security* to specify the number of layers used to contain the symbol's information in the range 1-4. Error correction data fills unused space in the specified number of layers.
Full range – as for compact mode but in the range 1-32.
Security identifies the percentage of the symbol to contain error checking data. The range is determined by the mode. A value of 0% will produce the symbol with the default amount of error correction (i.e. 23% + 3 codewords). The higher the security level the greater the number of layers required to contain the symbol and thus its overall size.

Font scaling barcodes

Font scaling barcodes provide a number of scalability options, but are only applicable to publications intended for Postscript and PDF output.

Using font scaling barcodes with output datastreams other than Postscript or PDF will produce unpredictable results.

Font scaling barcode settings

Label

Enter an optional name for the barcode object that will appear in the logic map.

Value

Enter the value you want to use for the barcode. See “Adding values to objects” on page 104 for more information.

Mode

Font scaling mode barcodes can only be used with PDF and Postscript output devices and enable you to change the scale settings. For *Standard barcodes* see page 147 and for *Line drawing barcodes* see page 151.

(continued)	**Font scaling barcode settings**
Barcode group	Select *Universal types* or one of the 2D barcode groups to populate the *Type* list.
Type	If *Universal types* has been selected, this list is populated with font scaling barcodes, otherwise with the 2D barcodes available for the selected *barcode group*.
Options	These options are used with certain font scaling barcodes. The enabling of options is barcode type dependent. *Optional Check Digit* automatically places a check digit, also known as a checksum, within the barcode. The check digit is used to verify that the value of the barcode has been entered correctly. *Wide bars* can be used to increase the width of individual bars within a barcode.
Scale	These options are used to set the size of the barcode. *Height/width ratio* is used to set the height to width ratio of the codewords used in each row of the barcode. *Weight reduction* enables the font weight of the barcode to be changed. The *Point Size* option enables the size of the barcode font to be set independently of the point size of the paragraph the barcode is associated with.
Presentation	These are cosmetic options that affect the presentation of the barcode when it is included on the page. Select *90 degree orientation* to rotate the barcode 90 degrees clockwise. Use the *Color* option to change the color of the barcode from the default black.

Specific barcode options

PDF417

Code page – if you need to include non-Latin characters in the barcode this allows you to select an alternative code page.
Security defines the amount of additional code words in the final barcode which can be used to reconstruct the barcode if the barcode has been damaged or misread.
Barcode aspect is the height to width ration of the number of codewords per row. 0 will provide a tall, thin barcode and 100 will provide a short, wide barcode.
Start mode indicates the type of data most likely to occur at the start of the data to be encoded.
Font aspect controls the aspect of the font to be used to present the barcode.

Datamatrix

Code page – if you need to include non-Latin characters in the barcode this allows you to select an alternative code page.
Mode determines the shape of the final barcode.
Security defines the amount of additional code words in the final barcode which can be used to reconstruct the barcode if the barcode has been defaced or misread.

(continued)	**Font scaling barcode settings**
QR Code	*Code page* – if you need to include non-Latin characters in the barcode this allows you to select an alternative code page. This is only available when the mode is set to *binary*. *Mode* determines which type of data will be used with the barcode. *Security* defines the amount of additional code words in the final barcode which can be used to reconstruct the barcode if the barcode has been defaced or misread.
2 of 5	*IATA 2 of 5* creates a variation of the 2 of 5 barcode which is commonly used in the airline industry
Code 128 & UCC/EAN 128	*Trim spaces* removes any extra spaces included in the value used for the barcode. *No control code* removes the ability for the barcode type to be set to either A, B or C dynamically based on the optimal mix of alpha numeric characters.
Code 39	*Start/stop chars as asterisk* enables you to use an '*' symbol instead of an alphanumeric character for the start and stop characters of the barcode. *Inter-cipher gap (1 unit)* limits the gap between ciphers in the barcode to one unit. If this option is not selected the gap between ciphers may be inconsistent.
German Postal	*Post check digit* is only valid when the Auto Check Digit has also been selected. This option enables the barcode image to be used as Identcode and Leitcode symbols.
ISBN	*ISBN-13* will prefix the barcode with '978' to conform to new standards being introduced by ISBN agencies.
Italian Postal 2/5	*Insert Space* places a space between the barcode value and the check digit; otherwise a dash is used when a check digit is included.
ITF	*Vertical bearer bars* provide an outline around the bars of the barcode. *H gauge* adds an additional 'h' character to the barcode to check for impression depth and ink spread. This is for use with older printers. Additional light margin space is also provided by this option.
MSI	*Modulo 10 check digit* adds another mod 10 check digit to the barcode in addition to the standard single mod 10 check digit. *Modulo 11 check digit* places a mod 11 check digit before the normal mod 10 check digit.
Plessey	*Separate data from check digit* automatically places a space between the value of the barcode and the check digit.
Swiss Postal	*B-Post symbol* is selected if the barcode is to be used for Swiss B-Post instead of the default A-Post.
Telepen	*ESC character* makes the first character in the barcode an ASCII escape character to conform with certain telepen numeric systems. *Even length* will prefix the string of characters with a '0' where necessary to ensure the barcode value contains an even number of characters.

(continued) **Font scaling barcode settings**

Aztec

Mode determines the characteristics of the final barcode. Choose from:
Normal – allows you to set *Security* to any percentage in the range 1-99.
Compact – use *Security* to specify the number of layers in the range 1-4 used to contain the symbol’s information. Error correction data will fill unused space in the specified number of layers.
Full range – as for compact mode but in the range 1-32.
Security identifies the percentage of the symbol to contain error checking data. The range is determined by the mode. A value of 0% will produce the symbol with the default amount of error correction. The higher the security level the greater the number of layers required to contain the symbol and thus its overall size.

Line drawing barcodes

These barcodes are for use with AFP, PCL, Postscript and VPS output devices.

Line drawing barcode options

Label

Enter an optional name for the barcode object that will appear in the logic map.

Value

Enter the value you want to use for the barcode. See “Adding values to objects” on page 104 for more information.

Mode

Line drawing mode barcodes can only be used with AFP output devices and enable you to size the height and width of the bars that form the barcode. For *Standard barcodes* see page 147 and for *Font scaling barcodes* see page 148.

Barcode group

Select *Universal types* or one of the 2D barcode groups to populate the *Type* list.

Type

If *Universal types* has been selected, this list is populated with barcodes that are compatible with the output devices, otherwise with the 2D barcodes available for the selected *barcode group*.

Scale

These options are used to set the size of the barcode. Depending on which barcode you select, different scaling options will be available.

Presentation

These are cosmetic options that affect the presentation of the barcode when it is included on the page. Select *90 degree orientation* to rotate the barcode 90 degrees clockwise. Use the *Color* option to change the color of the barcode from the default black.

(continued) **Line drawing barcode options**

Specific barcode options

PDF417 options

Barcode aspect is the height to width ratio of the number of code words per row. 0 will provide a tall, thin barcode and 100 will provide a short, wide barcode. *Security* defines the amount of additional code words in the final barcode which can be used to reconstruct the barcode if the barcode has been damaged or misread. *Height of cell* and *Width of cell* define the size of the individual bits that make up the barcode.

Datamatrix options

Mode determines the shape of the final barcode. *Security* defines the amount of additional codewords in the final barcode which can be used to reconstruct the barcode if the barcode has been defaced or misread. *Width of single cell* defines the height and the width of the individual bits that make up the barcode.

Previewing barcodes

You can choose to preview a barcode to verify that it has been configured correctly and to see how it will be presented in the document.

ⓘ Only font scaling and Line Drawing barcodes can be previewed. The option is disabled for standard barcodes.

▶ **To preview a barcode:** configure the barcode as required in the *Barcode* dialog box and click *Preview*. If it contains an invalid value a warning message box is displayed, otherwise you will be able to view the barcode.

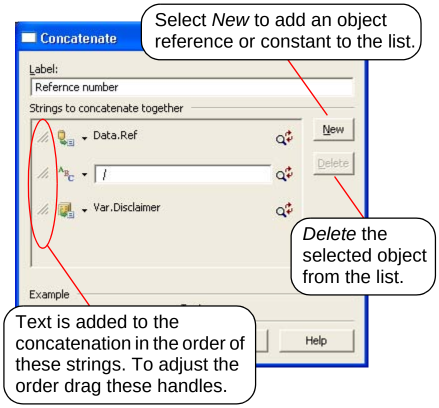
Concatenation

The concatenation function enables you to join together text values from two or more objects to form a single string. The function will treat all values passed to it as text regardless of the actual object type suppling it.

▶ To create a concatenation object:

1. Select *Insert/Field/Concatenation* from the menu
   – or –
   when specifying a value for an object, select the *Value* button and select *Functions/Concatenate*.

2. In the *Concatenation* dialog box click *New* to add the values to be concatenated. See “Adding values to objects” on page 104 for details.



Label: Refernce number

Data.Ref

/

Var.Disclaimer

Select *New* to add an object reference or constant to the list.

Delete the selected object from the list.

Text is added to the concatenation in the order of these strings. To adjust the order drag these handles.

Lookup table

A lookup table function allows you to replace predefined keywords with text strings or RTF fields at production run-time. The input keyword is checked against the entries in one or more lookup table files assigned to the job. If a match is found then the text in the lookup table is used to replace the keyword wherever this is used in publication logic.

> ⓘ NOTE THAT IF NO MATCH IS FOUND A SPECIAL CODE IS USED TO REPLACE THE STRING SO THAT IT CAN BE IDENTIFIED WITHIN THE OUTPUT DATASTREAM IF REQUIRED.

A lookup table is normally a simple text file. The keywords are first, followed by a space, followed by the text to be inserted. The keywords must not contain spaces and are case sensitive.

In this example, if the value assigned to the lookup table function is “sc2” then the string “Jackpot Bonus Scheme” will be inserted.

Keywords   Text to be inserted

```
sc1-t Winter Winners Plan
sc4-mx Spring Specials
sc2 Jackpot Bonus Scheme
```

For jobs intended to run under z/OS a lookup table can alternatively be created as a Key Sequenced VSAM dataset (KSDS). The keywords are provided as the VSAM key field and the substitution text is any other content in each record.

The files to be used as lookup tables for a production job are specified as part of a publishing task. You can override the lookup tables by using an Override Production Settings file (OPS). For more details, see “Running DOC1 Generate” on page 348.

▶ To create a table lookup function:

1. From the menu select *Insert/Field/Lookup Table* and proceed to step 3
   – or –
   create an object to receive the output.

   You can update a string variable with the function output or use it as input to any other object that requires a string value. See “Working with values” on page 104 for details.

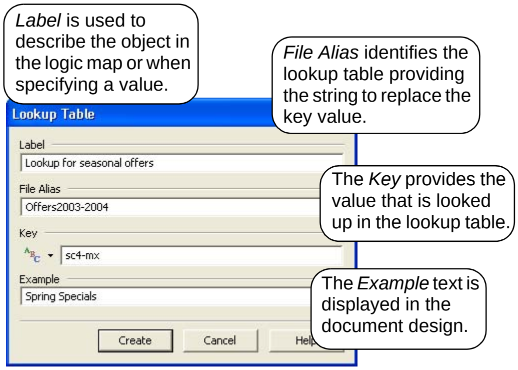
2. When specifying a value for the object select *Functions/Lookup table* as the type.

3. In the *Lookup Table* dialog box:

   Enter a *File Alias* that indicates the lookup table that should be used with this function.

   The *Key* provides the value that will be looked up. This must be text or convertible to text.

   The *Example* string entered is used within the document design itself.



Label: Lookup for seasonal offers
File Alias: Offers2003-2004
Key: sc4-mx
Example: Spring Specials

Label is used to describe the object in the logic map or when specifying a value.

File Alias identifies the lookup table providing the string to replace the key value.

The *Key* provides the value that is looked up in the lookup table.

The *Example* text is displayed in the document design.

Substring

The substring function extracts one or more consecutive characters from within a text string.

▶ To create a substring:

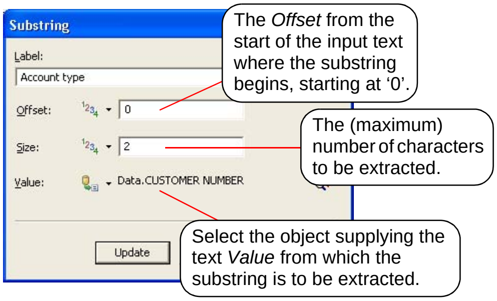
1. From the *Insert* menu select *Field/Substring* and proceed to step 3
   – or –
   create an object to receive the output.

   You can update a string variable with the function output or use it as input to any other object that requires a string value. See “Working with values” on page 104 for details.

2. When specifying a value for the object select *Functions/Substring* as the type.

3. In the *Substring* dialog box specify the input string and indicate the characters you want to extract.

   Note that if the offset is larger than the input text, an empty string is returned.



Substring

Label: Account type

Offset: 0

Size: 2

Value: Data.CUSTOMER NUMBER

The *Offset* from the start of the input text where the substring begins, starting at ‘0’.

The (maximum) number of characters to be extracted.

Select the object supplying the text *Value* from which the substring is to be extracted.

System variables

System variables allow you use run-time statistics and information from the production environment within functions and other objects; for example, you can input the number of pages generated by the current publication into an arithmetic or use the date when the production job was run with a barcode object.

System variables are also often used as input to a DOC1 journal to provide an index into the documents and pages within an output datastream generated by a production job.

▶ **To use a system variable:**

1. Create an object to receive the value.

   You can update a variable with the system variable value or use it as input to any other object that requires a value. See “Working with values” on page 104 for details.

2. When specifying a value for the object select *System Variable* as the type.

3. In the *System Variable* dialog box select the variable you want to use from the list.

System variables	
Document page number	The current page number within the document being processed.
Document vector	The offset from the start of the output datastream being created by DOC1 Generate to the start of the data that makes up the document currently being processed. The nature of this offset is system dependent – see “Vector offsets” on page 157 for details. This option is only available for use within Journal objects. To ensure that the offset is the correct one for the document you must place the journal object that sets the vector offset either within the document logic, or before the document in the publication logic.

(continued)	**System variables**
Elapsed CPU ticks	This measures the elapsed time in CPU milliseconds the calling process – such as DOC1 Generate or the *Refresh* option in the Document Editor – has taken up to the point in the logic map where the system variable is inserted. Placing the variable at multiple points in the logic map in conjunction with the *Print Message* action can be used to identify inefficient logic in the design of your DOC1 application. Note that while the time is measured in milliseconds it is output as a number.
Generation date/time	The system date or time when the HIP file being used by DOC1 Generate was published.
Job page number	The current page number of the entire production job, starting from 1.
Mandatory messages	This variable is only relevant for applications that include message objects created using Message1 and Content Author – see "Interfacing with message applications" on page 191 for more information. Returns 1 if there are messages that were marked as mandatory in the message application and are intended for the specified message area which have yet to be placed in the current document; otherwise returns 0.
Optional messages	As above except that 1 is returned if there are non-mandatory messages yet to be placed.
Page Vector	As for Document Vector but this will be the offset of the current page within the output datastream.
Publication page count	The total number of pages produced by the publication being processed. This option is only available for use within Journal objects.
Publication page number	The current page number within the publication being processed.
Publication sequence number	The number of publications that have been processed so far including the current publication.
Run date/time	The system date or time when DOC1 Generate began processing the production job.
Table column overflow	Only available within table headers and footers. Returns a *True* (1) value if the contents of a table flow to another presentation column or page, otherwise it returns a *False* (0) value. For more details, see "Using overflow variables" on page 69.
Table page overflow	As above but *True* or *False* is set only when a page overflow occurs.
Valid document flag	This value is set to 0 (zero) if there is an error with the publication currently being processed.

Note that some system variables are only available in certain instances, for example, the *Table* options can only be inserted in a table. For variables reporting page numbers be aware that these always relate to the number of 'logical pages' produced by the DOC1 production job and not necessarily the number of sheets produced on the printer/browser. For example, you may have chosen to place multiple logical pages on each sheet or have an output device that always adds separator pages to each job.

Vector offsets

The actual values generated for the Vector system variable depend on whether the output datastream is being generated as stream or record based data.

Record data is generated for AFP, Metacode and line data output under z/OS and for line data under OS/400. The vector value is the record count from the start of file and the first record is number 1.

Stream data is generated for all other output/host configurations. The vector value is the byte offset from the start of file. The first byte is number 0 (zero). Note that the values generated for document or page number '1' will not necessarily be the lowest possible count values. This is because protocols can contain header information related to the overall file but not directly relevant to a particular page.

Note that vector offset values are limited to ten digit numbers. As a result, where DOC1 Generate needs to deal with file offsets >9,999,999,999 it will use a base-32 encoding scheme to present the value. While the offset value will only be machine readable in these circumstances, you will be able to identify when this occurs as all values will be prefixed with 'Z'. For post-processing requirements PCE is able to read such values as a normal vector offset.

Keyed objects

DOC1 allows you to select some types of presentation objects dynamically according to variable criteria or 'keys'. Images and Active Content are supported in this way.

This can have a wide range of uses particularly where you would otherwise need to use many case or condition objects to select content. For example, you may want to use a branch number or similar to identify which of a large range of signatures or logo images needs to be placed. If you have more complex requirements you can use the same concept with Active Content objects that can themselves include presentation objects and associated logic.

NOTE THAT PARAMETERS AND RETURN VALUES ARE NOT SUPPORTED FOR KEYED ACTIVE CONTENT.

For keyed images, the simplest way to work is to import the files into the DOC1 Work Center so that their attributes are known and they can be referenced directly.

Where it is not possible to use integrated resources (i.e those directly available to the Designer) you can also reference external image resources – using one of two methods. You can create an external key map file that provides a reference to external images along with the relevant attributes. You can also create a user exit program that directly returns the image to be placed as a keyed object.

AN EXTERNAL KEY MAP IS CODED AS AN XML FILE WITH A PREDEFINED CONSTRUCT. THE STRUCTURE OF THIS FILE AND TOOLS TO HELP YOU CREATE IT ARE DOCUMENTED IN THE DOC1 PROGRAMMER'S GUIDE. THIS GUIDE ALSO CONTAINS FULL DETAILS FOR CREATING AND WORKING WITH USER EXIT PROGRAMS IN DOC1.

About key maps

The key map is central to using the keyed object feature. When using integrated resources the key map is a list of image or Active Content files that can be selected by a keyed object function. For external resources the key map allows you to include some sample content for use with the design itself and also provides a name by which the map is cross-referenced to the external control (file or user exit) at publication time.

Such key maps are then referenced when you actually place a *keyed object* within a design along with the *key value* itself which dictates which object is actually selected.

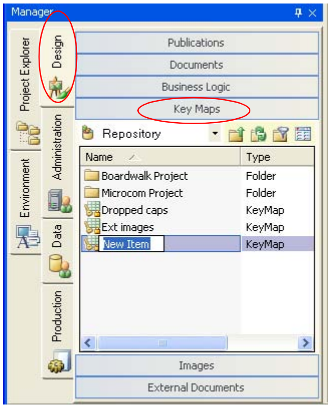
Key maps are maintained as individual files within the Work Center Manager. Once issued they become available for use with any publication in the repository. You cannot mix the type of object in a key map, but you can create any number of maps.

Objects do not already have to exist in the repository before being added to the key map; they can be created or imported as part of the add process.

▶ To create a key map:

1. In the Work Center Manager, select the *Design* tab and then click the *Key Maps* bar.
2. From the File menu select *New Key Map* and type in a name.
3. In the Key Map window use the *Type* field to specify the type of object to be referenced by this file.

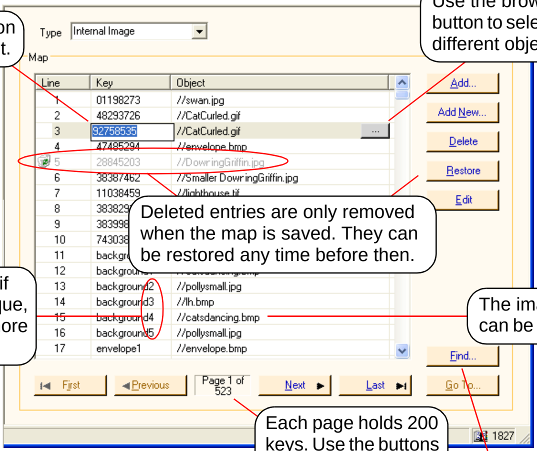
   You can now add entries to the key map.



Name	Type
Boardwalk Project	Folder
Microcom Project	Folder
Dropped caps	KeyMap
Ext images	KeyMap
New Item	KeyMap

▶ To add entries to a key map: in the Key Map window select the *Add* button. In the *Add* dialog box, enter a key value (i.e. the value that must be matched for the resource to be used) and identify the actual object that will be used. Note that new entries are inserted at the end of the map.

> ⓘ FOR EXTERNAL IMAGES, USING A REPRESENTATIVE SAMPLE OF THE ACTUAL IMAGES THAT WILL BE USED WILL HELP WHEN DESIGNING THE LAYOUT OF THE PAGE.



Line	Key	Object
1	01198273	//swan.jpg
2	48293726	//CatCurled.gif
3	92758535	//CatCurled.gif
4	47485294	//envelope.bmp
5	28845203	//DowringGriffin.jpg
6	38387462	//Smaller DowringGriffin.jpg
7	11038459	//lighthouse.tif
8	383829	
9	383998	
10	743038	
11	backgr	
12	backgrou	
13	background2	//pollysmall.jpg
14	background3	//lh.bmp
15	background4	//catsdancing.bmp
16	background5	//pollysmall.jpg
17	envelope1	//envelope.bmp

Double-click on a key to edit it.

Use the browse button to select a different object.

Deleted entries are only removed when the map is saved. They can be restored any time before then.

A suffix is added if the key is not unique, or when adding more than one object.

The image file name can be used in the key.

Each page holds 200 keys. Use the buttons to navigate through the pages …

… or use *Find* to search for a key or object.

▶ To view the image in a key map: select it in the key map and use the *View* button. The image will be displayed in the workspace.

▶ **To create new Active Content files while working with a key map:** in the key map select the *Add New* button. In the *Add New Active Content* dialog box, enter a key value and select a project for the new Active Content files. You can create up to 100 at a time. The new files are empty when created so you must edit them to create their actual content and logic. See “Using Active Content” on page 224.

▶ **To import images while working with a key map:** in the key map select the *Add New* button. In the *Add New Images* dialog box, enter a value or use the image filename for the key, select a target project into which the new images will be imported and browse for images.

Creating the keyed object

Within a publication design keyed objects can be inserted anywhere that other graphics objects can be used. The object actually placed can be of variable size – other content will flow around it according to the attributes of the keyed object. The object cannot, however, be rotated.

If a keyed object that references integrated resources is used within a text box then the text box will grow as required to fit the content. When referencing external images however, the eventual size of the image is not known and you will therefore need to ensure that text boxes are sized to accommodate the largest keyed object. In this situation a runtime error will be generated if the text box overflows.

You can monitor whether a keyed object has actually been placed at production run-time by specifying a variable into which a status code will be returned. This can then be used within conditional logic to deal with any problems.

The status values are:

0 – no keyed object has yet been encountered

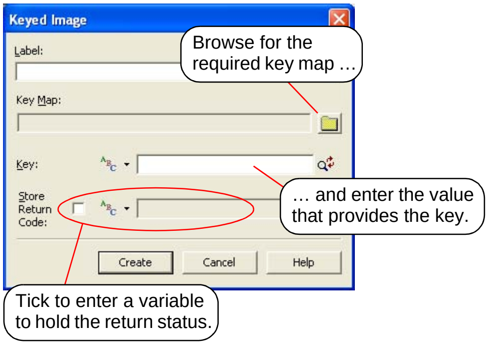
1 – object was placed successfully

2 – no object was found

3 – no object was found so the default object was placed instead (external keyed objects only)

▶ **To insert a keyed object:**

1. In the document editor logic map mark the insertion point as required and from the Insert menu select *Presentation objects/Keyed image* or *Keyed Active Content*.



Browse for the required key map …

… and enter the value that provides the key.

Tick to enter a variable to hold the return status.

2. Select the key map.

   In the *Keyed object* dialog box select the key map that references the objects you want to use. Note that only key maps that have already been issued will be available.

3. Select the key.

   This provides the value that identifies which object is used. Select a data field or other object that provides the variable data to match the keys specified in the key map. See “Working with values” on page 104.

Once inserted the editor will select the keyed object for display according to the current value of the key (for instance, the field value from the current document in sample data). When referencing external images the editor will show the appropriate object according to any sample entries in the key map (if any).

You can move keyed images and change their properties as for any shape – see “Working with graphics” on page 48. Double-click on the object and use the *Keyed object reference* dialog box to change to a different object.

▶ **To use a different key or key map in a document:** use the *Keyed image* tab in the *Image* dialog box.

Charts

DOC1 supports a range of charts that graphically represent data. The following types of charts are available in the DOC1 Work Center. Note that two-dimensional vertical and horizontal bar charts are always available; other types are only available subject to your license keycodes.

2-D bar chart – vertical and horizontal

3-D bar chart – vertical and horizontal

Stack chart (2-D & 3-D)

Line graph

Scatter chart

2-D pie chart

3-D pie chart

Chart terminology

Elements the individual elements that makes up a chart – a bar, a pie segment, a line, etc.

Chart data the values that are represented in the graphic.

Sets and Groups used by some chart type to categorize data of the same level. For instance, a stacked bar chart may represents year-on-year statistics for a number of products. A set would represent the years and the products would be referenced in a group.

Index used to cross reference chart elements with data values in some chart types.

Axes the lines on the outside of a chart that provide a scale reference.

Grid lines optional lines drawn across the chart at intervals on the scale.

Legend provides a key for chart elements.

Units used by some chart types to specify the scale to be used when representing values. For instance, 64000 would show as a chart value of 64 if units are set to 1000.

Charts are typically created using a wizard which assists with selecting the required chart configuration and which automatically creates the appropriate object tree in the logic map. A new chart created using the wizard has dummy values assigned to it. You will need to edit the *Chart data* objects it contains to supply the actual values you want to represent.

▶ To create a chart:

1. Position the cursor in the paragraph or other presentation object the chart is to follow.
2. On the *Insert* menu click *Presentation Objects/ Chart/ Chart* . The Chart wizard will be invoked.
3. Use the pages of the chart wizard to specify the initial setting for your chart.

   When you select *Finish* the new chart is generated and contains all the objects necessary to format the graphic according to your instructions.

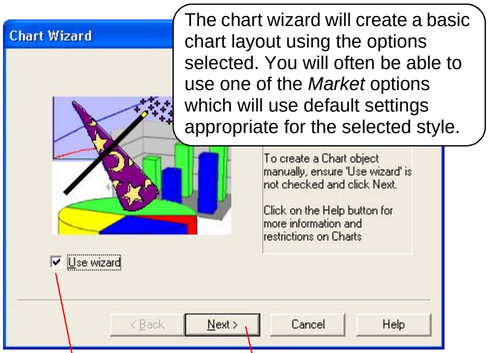


Chart Wizard

To create a Chart object manually, ensure 'Use wizard' is not checked and click Next.

Click on the Help button for more information and restrictions on Charts

☑ Use wizard

The chart wizard will create a basic chart layout using the options selected. You will often be able to use one of the *Market* options which will use default settings appropriate for the selected style.

Ensure that *Use wizard* is checked.

Select the *Next* button and follow the instructions on each screen.

4. Specify the values to be represented.

   In the logic map the main chart object will contain *Chart data* objects for each of the elements specified in the wizard. A dummy value (defined as a constant) will have been assigned to each.

   Double-click a *Chart data* object to show its properties dialog box. On the *Data* page use the *Value* control to select a data element or other object that will provide the chart input. See “Working with values” on page 104 for more information. You must only supply numeric values to a chart.

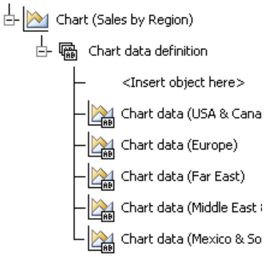


Chart (Sales by Region)
Chart data definition
<Insert object here>
Chart data (USA & Cana
Chart data (Europe)
Chart data (Far East)
Chart data (Middle East
Chart data (Mexico & So

5. Refine and customize the chart layout.

   Refer to the information about the chart type you are creating for details.

ⓘ IF REQUIRED YOU CAN MANUALLY GENERATE ALL THE OBJECTS REQUIRED TO BUILD A CHART. IF YOU WANT TO DO THIS CREATE THE MAIN CHART OBJECT AS ABOVE BUT UNCHECK THE *USE WIZARD* OPTION.

Moving and resizing

Charts are treated like any other graphic in the document editor and can be moved, resized and deleted as required. For more information about editing in general see “Working with graphics” on page 48.

Chart axes and scaling

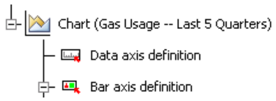
All charts except pie charts have two axes – the lines at the left and bottom of the graphic that set the scale for the values presented. The vertical line is always known as the *Data axis*. The name of the horizontal axis varies according to the chart type. The axes can have tick marks at user defined intervals, be labeled or unlabeled and you can restrict the data range to be displayed.

To enable the various tick mark settings, *Major Ticks* must be enabled. Once selected this allows you to choose the color for the ticks and whether you want to have minor ticks and labels in your chart.

Selecting the *Show grid lines* option will display grid lines in the chart and enable you to select the width and color of the lines.

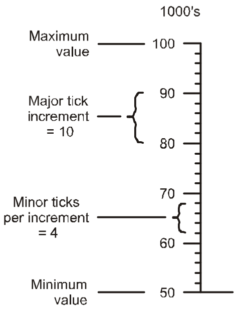
Units determine how the data presented in the chart is scaled to fit the axes. Bar and stack charts scale one axis; line and scatter charts scale both axes. For example, a bar chart with data axis units of 1000 will show a value of 83512 as a bar representing 84. Note that scaled values are always rounded up to the nearest whole number.

▶ **To customize axes and define units:** double-click the relevant axis object in the logic map. The object dialog box that is invoked will allow you to define the axis and unit settings that are relevant for the chart type and axis you are working with.



Axes and data values

The settings, *Maximum value*, *Minimum value*, *Major tick increment* and *Minor ticks per increment* together determine the look of a chart axis. The first three of these are expressed in terms of the *Units* specified; for example, if units is set to 1000 then a *Maximum value* of 8 actually indicates a maximum value of 8000.



Maximum value and *Minimum value* set the upper and lower limits of the values to be displayed. Values above the maximum will be cropped; those below the minimum will produce an 'empty' element. If these settings both have a value of 0 (zero) the chart scale will automatically be adjusted to fit the maximum and minimum values to be presented.

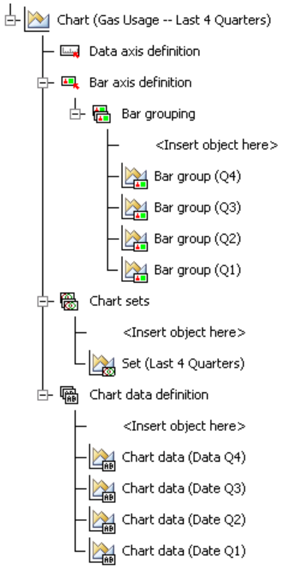
If your chart displays the actual data values for each element the correct values will be shown even if they are outside the 'visual' range set by the *Maximum/Minimum value* settings.

Major tick increment is the number of units between tick marks on the scales axis. These marks have their scaled value next to them. The *Minor ticks per increment* is the number of marks between each major tick. Note that both major and minor ticks are optional.

Sequencing and indexes

For scatter and pie charts the order in which *chart data* objects appear in the logic map automatically determines the sequence in which the elements will appear in the graphic.

Bar, stack and line charts work with groups of values and you will need to use index numbers to associate data objects with the appropriate group. Line chart elements have a single index number – one per line in the chart. Bar and stack chart elements have two index numbers – one for the *Group* and one for the *Set* to which they belong. You will need to ensure that all *Chart data* objects reference the appropriate index number(s).



```
Chart (Gas Usage -- Last 4 Quarters)
    Data axis definition
    Bar axis definition
        Bar grouping
                <Insert object here>
            Bar group (Q4)
            Bar group (Q3)
            Bar group (Q2)
            Bar group (Q1)
    Chart sets
            <Insert object here>
        Set (Last 4 Quarters)
    Chart data definition
            <Insert object here>
        Chart data (Data Q4)
        Chart data (Date Q3)
        Chart data (Date Q2)
        Chart data (Date Q1)
```

If you used the wizard to create your chart then the element and data objects will have been automatically created with the required index numbers to provide these references. You will need to use new indexes if you add elements to the chart.

In addition to providing cross references the index numbers also represent the sequence in which the elements will be added to the chart, starting from 1 and with a maximum of 32. You will normally want to ensure that the sequence has no gaps but note that, should a particular index number not be assigned a value, then no data will appear on the chart at that position.

You should refer to the information about specific chart types for details of creating and referencing indexes.

Bar and stack charts

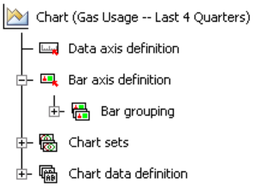
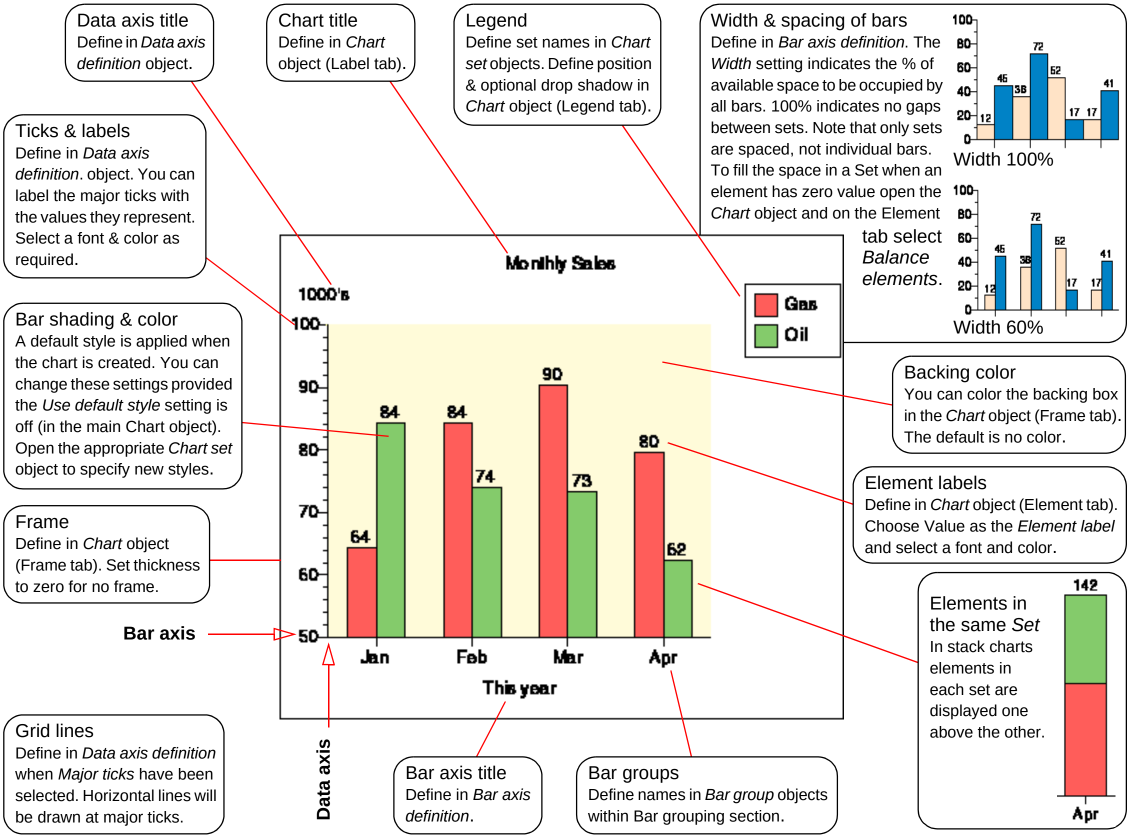


Chart (Gas Usage -- Last 4 Quarters)
- Data axis definition
- Bar axis definition
  - Bar grouping
- Chart sets
- Chart data definition

Bar charts represent data as 2 or 3-dimensional, vertical or horizontal bars. You can group data elements together in bars that are placed side by side. Stack charts are similar to bar charts except that grouped data elements are placed one above the other in a single bar.

Values within bar or stack charts are always part of a *Set* and a *Bar group*. The set identifies values of the same type: for example, the same month in a number of years. The bar group places values within a comparison group: for example, a particular year. If your chart has only one comparison to make – for example, only months for a single year are to be shown – then the group will always be the same.



Data axis title
Define in *Data axis definition* object.

Chart title
Define in *Chart* object (Label tab).

Legend
Define set names in *Chart set* objects. Define position & optional drop shadow in *Chart* object (Legend tab).

Width & spacing of bars
Define in *Bar axis definition*. The *Width* setting indicates the % of available space to be occupied by all bars. 100% indicates no gaps between sets. Note that only sets are spaced, not individual bars. To fill the space in a Set when an element has zero value open the *Chart* object and on the Element tab select *Balance elements*.

Ticks & labels
Define in *Data axis definition*. object. You can label the major ticks with the values they represent. Select a font & color as required.

Bar shading & color
A default style is applied when the chart is created. You can change these settings provided the *Use default style* setting is off (in the main Chart object). Open the appropriate *Chart set* object to specify new styles.

Frame
Define in *Chart* object (Frame tab). Set thickness to zero for no frame.

Backing color
You can color the backing box in the *Chart* object (Frame tab). The default is no color.

Element labels
Define in *Chart* object (Element tab). Choose Value as the *Element label* and select a font and color.

Elements in the same *Set*
In stack charts elements in each set are displayed one above the other.

Grid lines
Define in *Data axis definition* when *Major ticks* have been selected. Horizontal lines will be drawn at major ticks.

Bar axis title
Define in *Bar axis definition*.

Bar groups
Define names in *Bar group* objects within Bar grouping section.

See “Sequencing and indexes” on page 165 for details

Sets and groups are defined as independent objects within the main chart object and must be assigned a unique index number along with their text description that is actually used on the chart. The chart data objects that define the values to be represented on the chart must reference these indexes.

▶ **To create a new Bar group:** mark the insertion point within the Bar grouping section of the Chart object in the logic map and from the *Insert* menu click *Presentation Objects/Chart/Chart Group*.

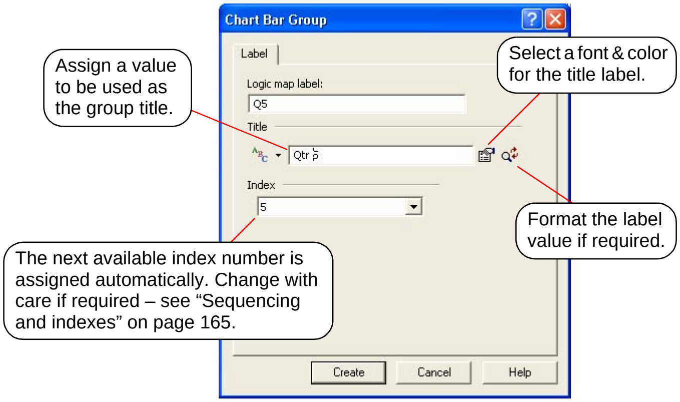


Chart Bar Group

Logic map label: Q5

Title: Qtr $

Index: 5

Assign a value to be used as the group title.

Select a font & color for the title label.

Format the label value if required.

The next available index number is assigned automatically. Change with care if required – see “Sequencing and indexes” on page 165.

▶ **To create a new Set:** mark the insertion point within the Chart sets section of the Chart object in the logic map and from the *Insert* menu click *Presentation Objects/Chart/Chart Set*.

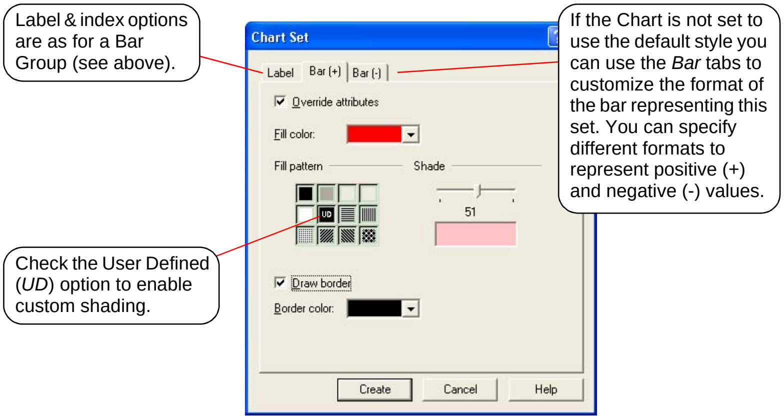


Chart Set

☑ Override attributes

Fill color: 🟥

Fill pattern: UD

Shade: 51

☑ Draw border

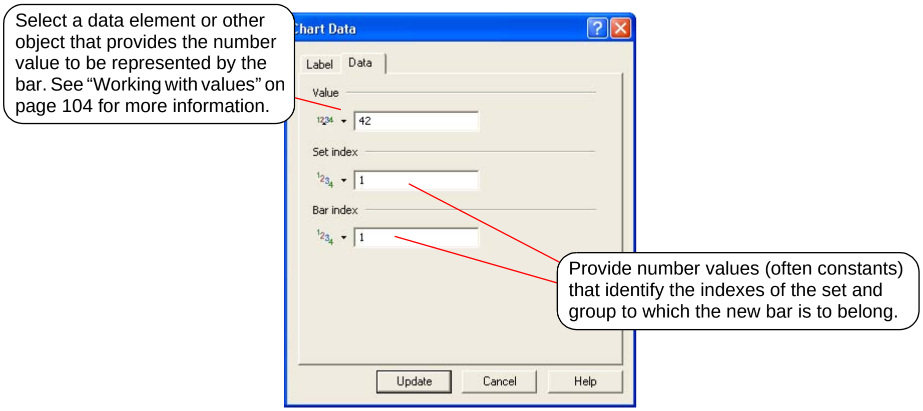
Border color: ■

Label & index options are as for a Bar Group (see above).

If the Chart is not set to use the default style you can use the *Bar* tabs to customize the format of the bar representing this set. You can specify different formats to represent positive (+) and negative (-) values.

Check the User Defined (*UD*) option to enable custom shading.

▶ **To add a new bar:** open the Chart data definition section within the Chart object in the logic map and from the *Insert* menu click *Presentation Objects/Chart/Chart Data*.



Select a data element or other object that provides the number value to be represented by the bar. See “Working with values” on page 104 for more information.

Chart Data

Label | Data

Value: 42

Set index: 1

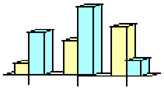
Bar index: 1

Provide number values (often constants) that identify the indexes of the set and group to which the new bar is to belong.

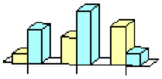
Tilting

You can tilt a 3-dimensional chart to create a greater 3-D effect. You specify the angle that the chart is tilted from the horizontal. 3-D charts are created with a default tilt angle of 5°.

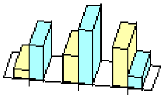
▶ **To tilt a 3-dimensional chart:** open the Chart object in the logic map and in the *View* tab set the *tilt* to an angle in the range 5-45°.



Tilt of 5%



Tilt of 15%



Tilt of 45%

Line charts

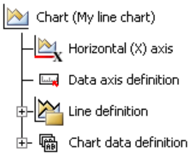
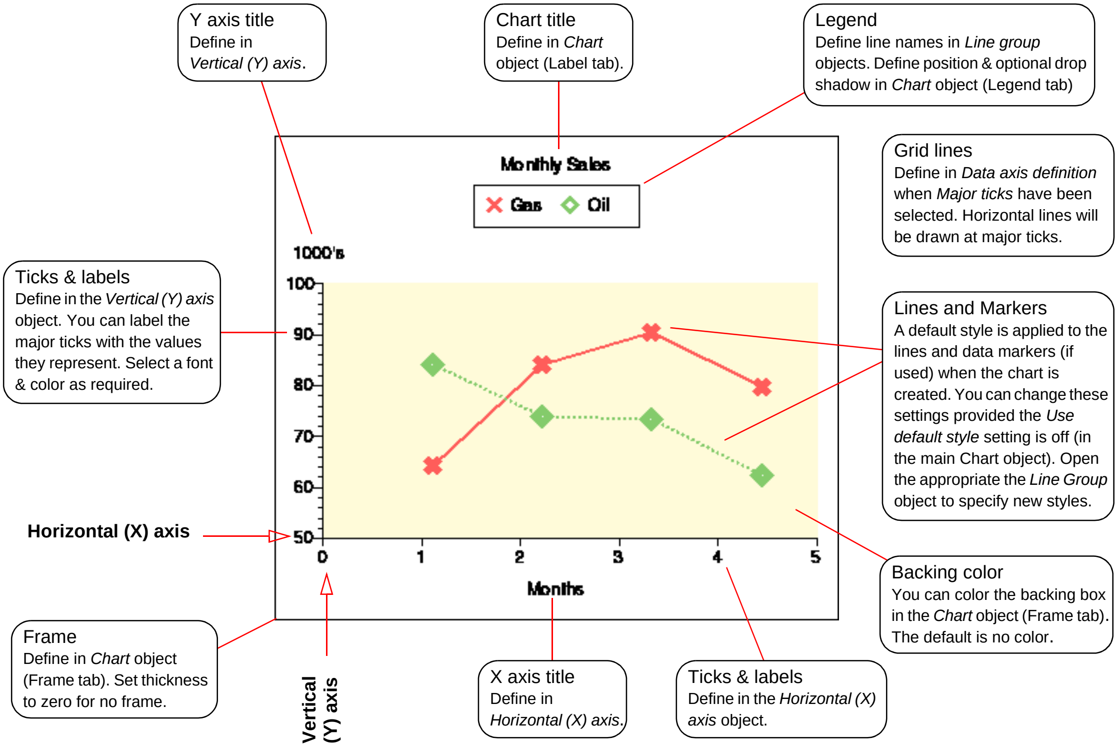


Chart (My line chart)
- Horizontal (X) axis
- Data axis definition
- Line definition
- Chart data definition

Line charts plot a series of X/Y coordinates along one or more lines. These coordinates are defined in the chart data objects assigned to the chart. The coordinates can be highlighted with markers if required.

Each line is defined as an independent object within the line definition section and must be assigned a unique index number along with a text description to be used in the chart legend. The chart data objects that define the values to be represented on the chart must reference these indexes to be associated with a particular line.



Y axis title
Define in *Vertical (Y) axis*.

Chart title
Define in *Chart* object (Label tab).

Legend
Define line names in *Line group* objects. Define position & optional drop shadow in *Chart* object (Legend tab)

Grid lines
Define in *Data axis definition* when *Major ticks* have been selected. Horizontal lines will be drawn at major ticks.

Ticks & labels
Define in the *Vertical (Y) axis* object. You can label the major ticks with the values they represent. Select a font & color as required.

Lines and Markers
A default style is applied to the lines and data markers (if used) when the chart is created. You can change these settings provided the *Use default style* setting is off (in the main Chart object). Open the appropriate the *Line Group* object to specify new styles.

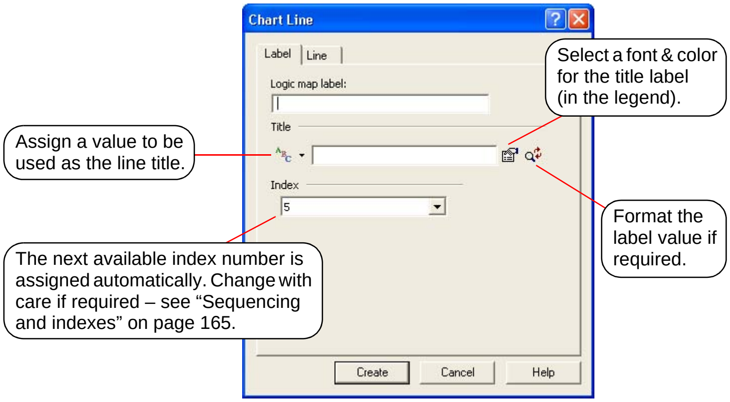
Backing color
You can color the backing box in the *Chart* object (Frame tab). The default is no color.

Frame
Define in *Chart* object (Frame tab). Set thickness to zero for no frame.

X axis title
Define in *Horizontal (X) axis*.

Ticks & labels
Define in the *Horizontal (X) axis* object.

▶ **To create a new line:** mark the insertion point within the Line definition section of the Chart object in the logic map and from the *Insert* menu click *Presentation Objects/Chart/Chart Line*.



Logic map label:

Title

Index: 5

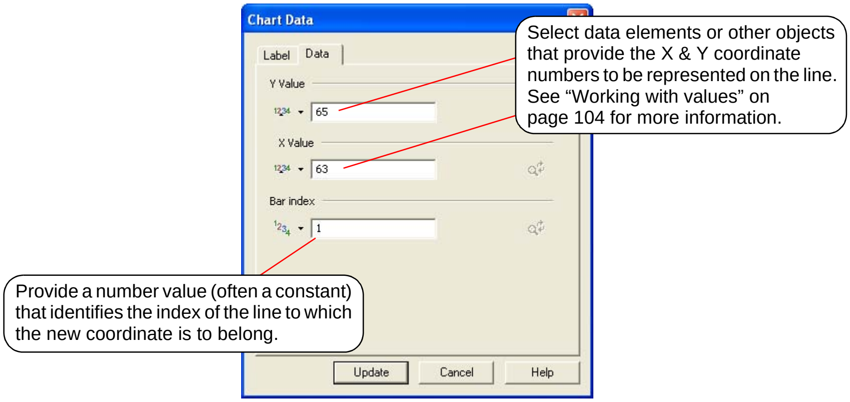
Assign a value to be used as the line title.

Select a font & color for the title label (in the legend).

Format the label value if required.

The next available index number is assigned automatically. Change with care if required – see “Sequencing and indexes” on page 165.

▶ **To add a new coordinate:** open the Chart data definition section within the Chart object in the logic map and from the *Insert* menu click *Presentation Objects/Chart/Chart data.*



Y Value: 65

X Value: 63

Bar index: 1

Select data elements or other objects that provide the X & Y coordinate numbers to be represented on the line. See “Working with values” on page 104 for more information.

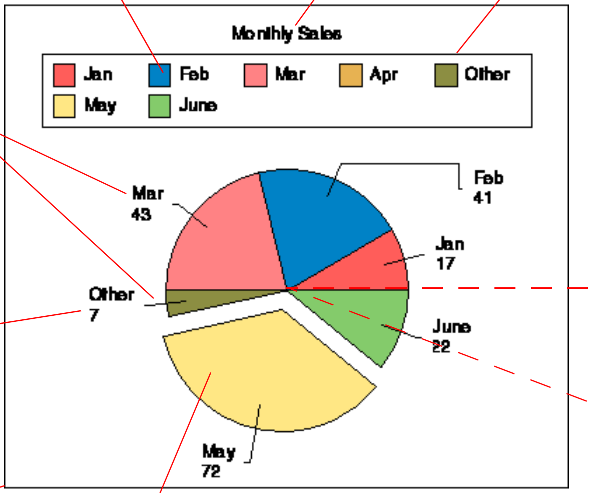
Provide a number value (often a constant) that identifies the index of the line to which the new coordinate is to belong.

Pie charts

A pie chart shows data as a circular diagram with segments indicating the percentage each value represents of the overall total. The pie itself can have a two or three-dimensional look. Two-dimensional charts have the additional option of 'exploding' either the largest or all of the segments.

Each chart data object assigned to the chart equates to a segment of pie that can appear in the graphic but note that you may choose to ignore or group values below a user-defined threshold. DOC1 will calculate the percentage to be assigned to each segment depending on the contents of all the chart data objects assigned.

Pie charts do not use index numbers. The pie segments are displayed in the sequence they are placed with the chart data definition group.



Segment shading & color
A default style is applied when the chart is created. You can change these settings provided the *Use default style* setting is off (in the main Chart object). Open the appropriate *Chart data* object to specify new styles.

Element labels and pointers
Define the type of label required in the *Chart* object (Element tab). You may optionally include the segment name and the data either as a percentage or the actual number.
A line will join the label to the middle of the appropriate segment. You can choose to align all labels to the left & right of the chart in the *Chart* object (Type tab).

Chart title
Define in *Chart* object (Label tab)

Legend
Define segment names in *Chart data* objects. Define position & optional drop shadow in *Chart* object (Legend tab).

Start position
By default the element relating to the first data object in the data definition group is placed with its bottom line at the 3 o'clock position.
Change this in the *Chart* object *View* tab by specifying a degree of rotation (0-360°).

3 o'clock

20° rotation would place the bottom of the first segment here.

Small values
You can group small values into one category with or without labels. See “Customizing small segment display” on page 173 for more details.

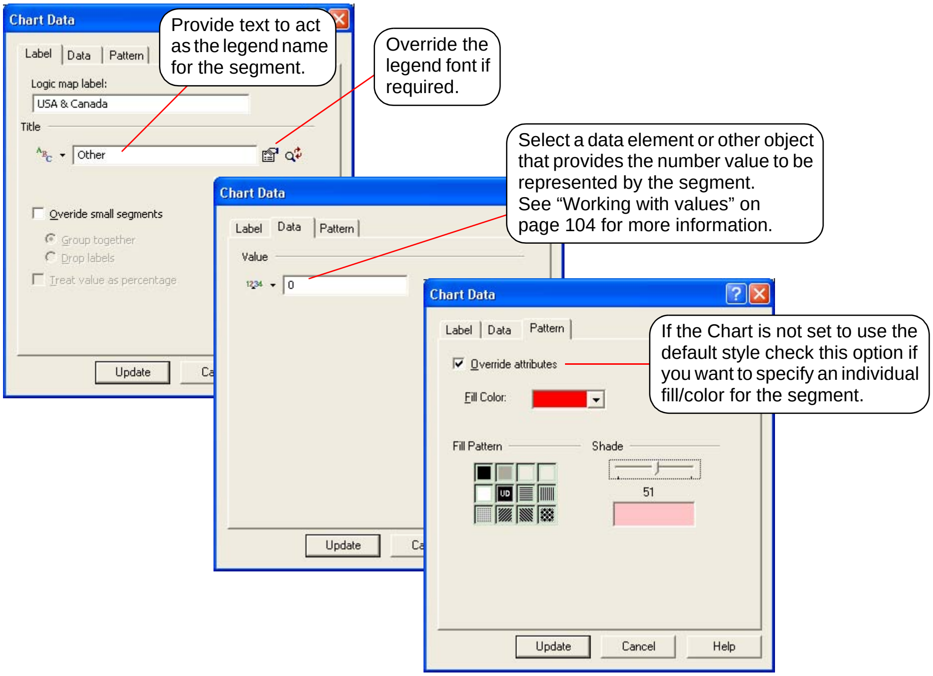
Frame
Define in *Chart* object (Frame tab). Set thickness to zero for no frame.

Exploded segment
Double-click on the *Chart* object and select the *View* tab. You can specify that all the segments are exploded or just the largest one.

Position/size of the pie
You can move and adjust the size of the pie relative to the overall chart frame. In the *Chart* dialog box (Frame tab), check *Move/resize the pie*. This enables the position and size options. Change as required.
Note that *Height* is only enabled for a 3D pie chart, and the position is measured from the top left corner of the frame.

▶ **To add a new segment:**

1. In the logic map open the Chart data definition section within the Chart object
2. Click on the existing chart data object that the new segment is to precede.
3. From the *Insert* menu click *Presentation Objects/Chart/Chart data.*



Logic map label: USA & Canada

Title: Other

☐ Overide small segments

○ Group together

○ Drop labels

☐ Treat value as percentage

Value: 0

☑ Override attributes

Fill Color: 🟥

Shade: 51

Provide text to act as the legend name for the segment.

Override the legend font if required.

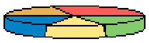
Select a data element or other object that provides the number value to be represented by the segment. See “Working with values” on page 104 for more information.

If the Chart is not set to use the default style check this option if you want to specify an individual fill/color for the segment.

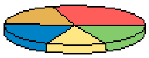
Tilting

You can tilt a 3-dimensional chart to enhance the 3-D effect. You specify the angle that the chart is tilted from the horizontal. 3-D charts are created with a default tilt angle of 5˚.

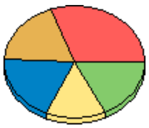
▶ **To tilt a 3-dimensional chart:** open the Chart object in the logic map and in the *View* tab set the *tilt* to an angle in the range 5-45˚.



Tilt of 5%



Tilt of 15%



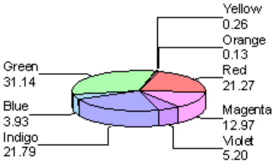
Tilt of 45%

Labels

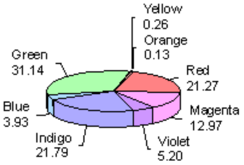
Element labels can be drawn either close to the pie segments or aligned on the right and left sides of the chart.

▶ **To change the position of labels on the chart:** open the Chart object in the logic map and in the *Type* tab use the *Align pie segment labels* option.

NOTE THAT LABEL POINTERS MAY BE COMBINED WHERE SMALL SEGMENTS OVERLAP.



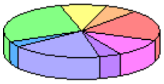
Labels aligned



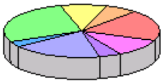
Labels not aligned

Color of 3-D pie chart base

The base of a three-dimensional pie chart can use either the same colors as the top of individual segments (this is the default), or you can specify a single color for the entire base.



Default shading



Specific shading color

▶ **To set shading for the pie base:** open the Chart object in the logic map and in the *View* tab check *Use shading*. Select the *fill color, pattern* and *shade* as required.

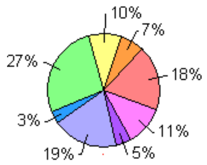
Customizing small segment display

Your chart may be assigned values that would generate only a very small pie segment and which may not look very clear. For this reason there are several options that allow you to customize how such values are handled. These are all based on the threshold value initially set in the Chart wizard or Chart data object that determines when a value is to be treated as 'small'.

By default, the first Chart data object in the logic map – labelled *Other* – contains the small segment options.

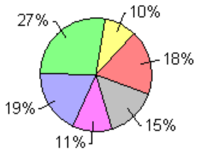
▶ **To set the threshold for small segments:** open the Chart data object in the logic map and in the *Data* tab use *Value* to set the number below which the small segment options will be applied. By default, the threshold value relates to actual chart data but if you select the *Treat value as percentage* option the value represents a percentage of the overall pie. If a value will produce a segment smaller than the percentage the small segment options will be applied.

▶ **To show all values regardless of size:** open the Chart data object in the logic map and ensure that the *Override small segments* option is not checked. All segments will appear in the chart, regardless of size.



▶ **To remove labels from small segments:** open the Chart data object in the logic map and ensure that *Override small segments* is checked and click *Drop labels*. The labels for segments with a value less than the threshold will not be displayed.

▶ **To group small values together in one segment:** open the Chart data object in the logic map and ensure that *Override small segments* is checked and select *Group together*. In *Pattern*, specify the color for the grouped segment and use *Title* to select title text for the group segment.



Scatter charts

Scatter charts plot a series of X/Y coordinates within the chart scale. The coordinates can be represented by markers of varying styles.

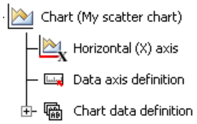


Chart (My scatter chart)
- Horizontal (X) axis
- Data axis definition
- Chart data definition

The coordinates are not connected to one another and therefore scatter charts do not use index numbers. However, you can use grid lines to assist with reading the relative scale position of the values provided.

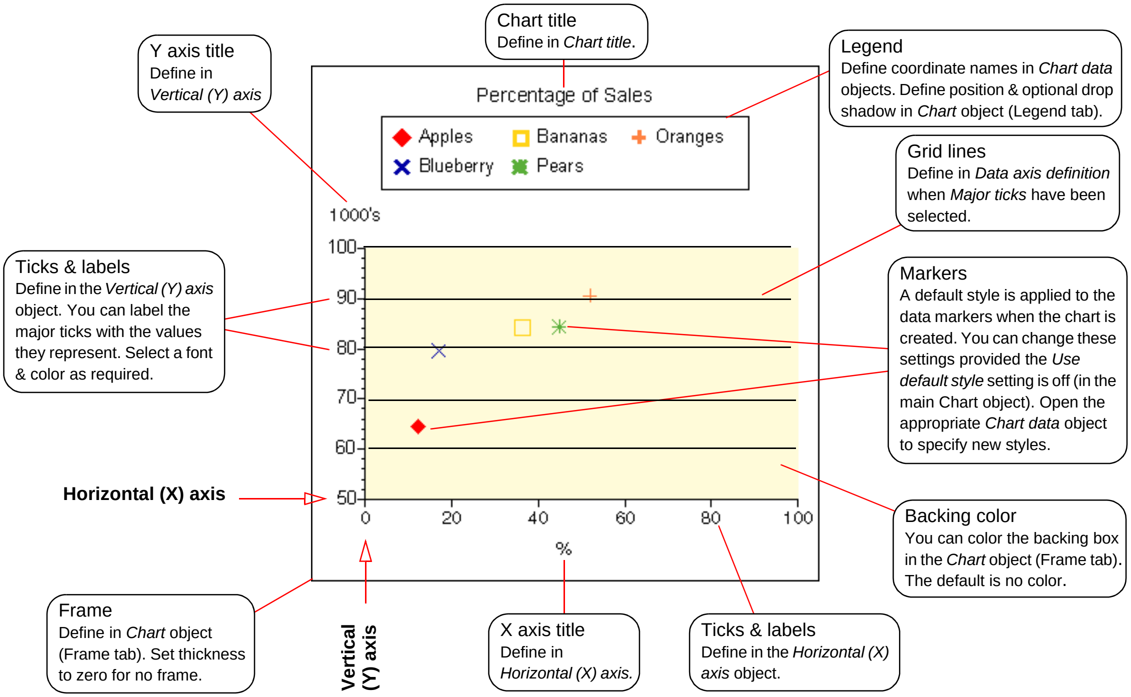


Chart title
Define in *Chart title*.

Y axis title
Define in *Vertical (Y) axis*

Legend
Define coordinate names in *Chart data* objects. Define position & optional drop shadow in *Chart* object (Legend tab).

Grid lines
Define in *Data axis definition* when *Major ticks* have been selected.

Ticks & labels
Define in the *Vertical (Y) axis* object. You can label the major ticks with the values they represent. Select a font & color as required.

Markers
A default style is applied to the data markers when the chart is created. You can change these settings provided the *Use default style* setting is off (in the main Chart object). Open the appropriate *Chart data* object to specify new styles.

Horizontal (X) axis

Backing color
You can color the backing box in the *Chart* object (Frame tab). The default is no color.

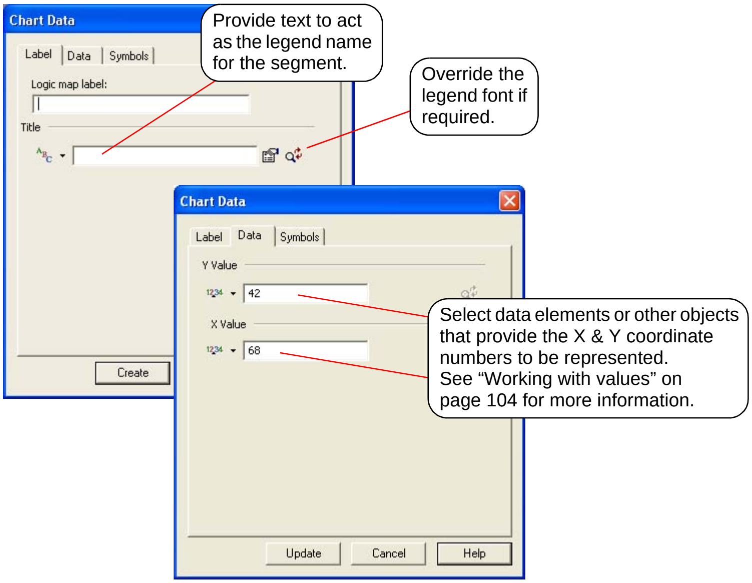
Frame
Define in *Chart* object (Frame tab). Set thickness to zero for no frame.

Vertical (Y) axis

X axis title
Define in *Horizontal (X) axis.*

Ticks & labels
Define in the *Horizontal (X) axis* object.

▶ **To add a new coordinate:** open the Chart data definition section within the Chart object in the logic map and from the *Insert* menu click *Presentation Objects/ Chart/ Chart data.*



Y Value: 42

X Value: 68

Provide text to act as the legend name for the segment.

Override the legend font if required.

Select data elements or other objects that provide the X & Y coordinate numbers to be represented. See “Working with values” on page 104 for more information.

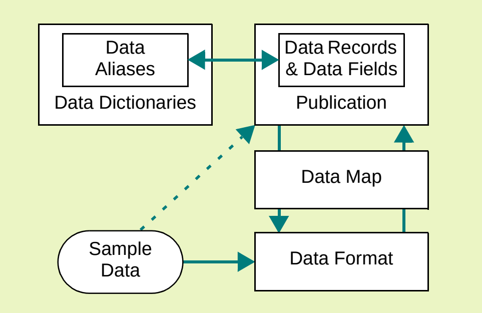
Advanced data handling

The information in the chapter “Working with input data” on page 72 details most of the features you will need to use when using elements from your intended input data file in publication designs. This section covers any other topics related to the use of input data not covered elsewhere.

Sample data and data mapping

You can create a data map as part of the publishing task itself. See “Publishing and deployment” on page 329 for details.

When you are designing publications you work with indirect references to the records and fields that will supplied by the input data file in the production environment. The actual data elements are defined in a data format file which uses a sample of the intended input data as a template. Before publishing the data aliases within a publication must be linked to the relevant elements in a data format using a data map object.



This allows a great deal of flexibility for creating and maintaining data references within designs but until a publication has been associated with a data format the document editor is unable to use variable examples to show data elements. Data fields are displayed using the example data you specified when they were created.

If you want to see the contents of a sample data file used in the document editor you will need to create a data map object. You can specifically create a new data map and associate it with a data format and your publication. You can then add individual links between the data elements defined in each.

Alternatively you can export both a data map and a data dictionary from an existing data format. This will allow designers to work with a list of known data elements and avoids the need to manually link individual elements.

Data maps and sample data can also be associated with components of a publication. While editing public documents and Active Content independently of a publication you can use data maps and sample data of your choice. This provides the flexibility to see how the design of a document or Active Content used in multiple publications is affected by different sample data. Note that the link between a data map and a publication is always the latest issued revision. If you make any changes these will not take effect until issued.

DATA MAP AND SAMPLE DATA ASSIGNMENTS MADE AT PUBLICATION LEVEL ARE PASSED DOWN TO ACTIVE CONTENT AND PUBLIC DOCUMENTS INCLUDED IN YOUR PUBLICATION. SEE "SHARING AND SYNCHRONIZATION" ON PAGE 101 FOR DETAILS.

▶ To associate a data map within a publication or change the sample data file to be used:

1. In the Publication View double-click the publication button
2. In the publication dialog use the *Data Map* browse button.

   You can select an existing data map or create a new one by selecting a data format. If a data format is selected the data mapping wizard will automatically be invoked.
3. If you want to change the sample data file associated with the data format/data map click the *Sample Data* browse button.

▶ To associate or edit a data map and sample data used by a publication:

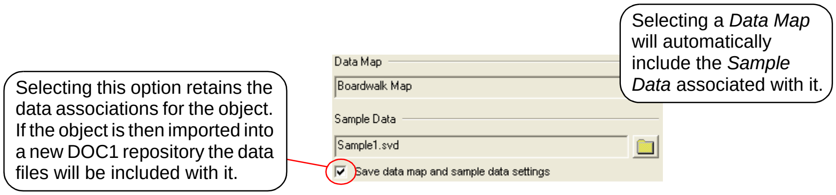
1. Select the Design tab and the Publications bar.
2. Highlight the publication and select *File/Properties* from the menu
   – or –
   right-click the publication and select *Properties* from the shortcut menu.
3. From the *Properties* dialog box select the *Data Map* tab.
4. Click the *Data Map* browse button and select a data map from the repository and click *OK*.
5. Click the *Sample Data* browse button and select a sample data file from the repository and click *OK*.

SEE "DATA MAPPING" ON PAGE 261 FOR MORE INFORMATION ABOUT DATA MAPS AND HOW TO CREATE THEM USING THE DATA MAPPING WIZARD.

▶ To associate a data map and sample data with a document or Active Content during editing:

1. Double-click on the document or Active Content object in the logic map and for Active Content only select the *Sample Data* tab.
2. Click the *Data Map* browse button and select a data map from the repository and click *OK*.
3. If necessary click the *Sample Data* browse button and select a different sample data file from the repository and click *OK*.

4. Select *Save data map and sample data settings* to retain those settings next time the file is opened.



Selecting this option retains the data associations for the object. If the object is then imported into a new DOC1 repository the data files will be included with it.

Selecting a *Data Map* will automatically include the *Sample Data* associated with it.

Data Map

Boardwalk Map

Sample Data

Sample1.svd

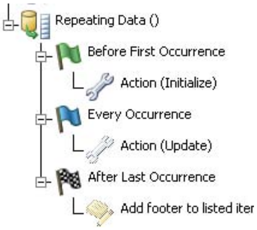
☑ Save data map and sample data settings

Using a repeating data control

The *repeating data* object allows you to process the multiple iterations of a repeating data element without using a transaction table. As with a transaction table you can choose to sort the data within the control object before it is actually processed within the function. You can also access the number of the current repeat and the total number of repeats during processing.

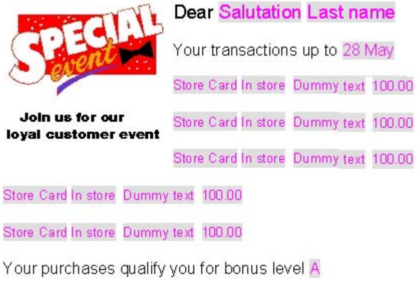
You may need to use this feature if you want to process repeating data without actually placing fields in a document design, for instance if you simply want to total values from the data and store them for later use. You may also find it allows you more flexibility than a transaction table if you need to create complex logic in relation to placing repeating data.

In the logic map the object has a main *Every Occurrence* group in which you specify the logic and, optionally, the presentation objects that will be processed iteratively according to the number of times the control element is repeated. It also has a *Before First Occurrence* group which is processed once when the control object is first encountered and an *After Last Occurrence* group which is processed once after all the repeating data has been handled. You can use presentation objects within the *Before* and *After* groups to create header and footer style sections around the repeating data if required – note that presentation objects should not be used when the repeating data object is in a table.



Repeating Data ()
Before First Occurrence
Action (Initialize)
Every Occurrence
Action (Update)
After Last Occurrence
Add footer to listed iter

All presentation objects within the object are placed within a single column format which can occupy all the available page width. Graphics and objects on the background may constrain the available space as during normal processing. Presentation objects within the *Every Occurrence* group are repeated for each iteration of the control object in the input data. These will automatically flow down the page and new pages will automatically be generated as required.


SPECIAL
event

Dear Salutation Last name

Your transactions up to 28 May

Store Card In store Dummy text 100.00

Join us for our loyal customer event

Store Card In store Dummy text 100.00

Store Card In store Dummy text 100.00

Store Card In store Dummy text 100.00

Store Card In store Dummy text 100.00

Your purchases qualify you for bonus level A

▶ To create a repeating data control:

1. In the logic map, click on the object after which the repeating data control is to be inserted.

2. From the *Insert* menu select *Control Objects/ Repeating Data*.

3. Specify the control for the repeating data.

   In the *Repeating Data* dialog box use the *Control Object* either to select the data record that is repeated or to specify the number of repeats in a variable.

   Note that you can use the *New Alias* button to create a new record if a data alias for the required element has not yet been established.

4. Specify data sorting.

   By default, repeating data will be presented according to the order in which it is provided in the input data file. If required, you can specify a sort that re-sequences the data prior to being placed in the document – this is available only when a data record has been selected as the repeat.

   ⓘ Sorting is based on the sequence of characters in the Unicode table. Results should be fully predictable when data contains standard ASCII-based characters but using sorting with other characters, particularly those from ideographic languages, may produce unexpected results.

   For a *Sort* the contents of a field within the repeating data is used as the sort key. Select the name of the appropriate data field in the *Field* control. You can also specify if the data is to be sorted ascending or descending.

5. Specify a repeat count.

   In the Advanced page of the *Repeating Data* dialog box you can use the *Monitor status using these variables* option to specify variables (or Active Content parameters) as required in which to save the number of the current repeat and the total number of repeats. These are standard variables, so once set, are always available for use.

6. Populate the *Before First Occurrence* group.

   Expand the Repeating Data object in the logic map. In the *Before First Occurrence* you can include one or more paragraphs that will be placed immediately above the repeating data itself. You may also want to add any logic to be executed before the repeating data is processed. Note that presentation objects should not be used when the repeating data object is in a table.

7. Add objects to be iterated in the body area.

   Click *Every Occurrence* within the group and add one or more paragraphs that will be iterated for each repeat. When using data records to control the repeat, you can include data fields from each record here. See “Using data fields within text” on page 73 for details. Note that presentation objects should not be used when the repeating data object is in a table.

   Consider using tabs within the paragraphs to line up fields if appropriate.

8. Populate the *After Last Occurrence* group.

   Paragraphs in this group will be placed immediately after the repeating data. You may also want to add any logic to be executed after the repeating data is processed.

Repeating data dialog box – General settings

Number of repeats to show in editor

This option controls how much example data for the repeating data control is shown in the document view. Where data fields have already been mapped to a data format the examples will be drawn from the sample data currently associated with the publication. Where data fields have yet to be mapped the example data specified for each field will be used (repeatedly if necessary). These options determine how many such sample transactions are displayed. IMPORTANT: this has influence only during editing and has no effect on how data is handled in preview or production.

Until new page — Enough transactions will be displayed so that all available space on the first page is filled plus one transaction on the new page.

All — If the publication has already been mapped to a data format, data from the currently active sample data file is used to control the repeats that are displayed.

Limit to a maximum of — Only the number of transactions specified will be displayed.

Spacing

These values indicate the amount of space to be left between the complete set of presentation objects produced by the repeating data object (if any) and the objects that are placed before and after. They do not influence space within the objects produced by the element itself. If you want to make these kind of adjustments use paragraph settings.

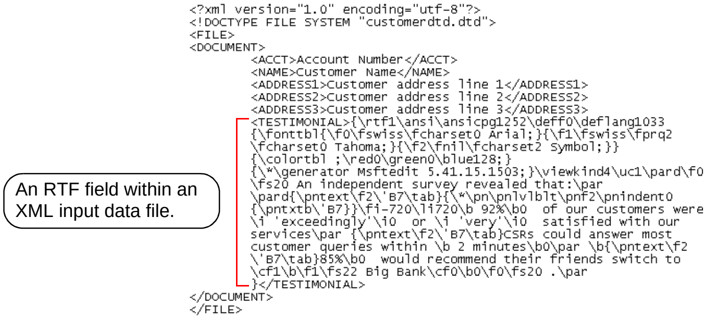
Including RTF fields in input data

DOC1 allows you to include RTF code within your input data and subsequently incorporate the formatted text directly into your design documents from the data format. The RTF can include simple text formatting but more advanced formatting is not supported – see "RTF" on page 299 for a list of restrictions.

XML input data must be used if you want to include hard returns in the RTF field. The RTF must be placed within a <CDATA> container for the hard returns to be handled correctly by DOC1GEN. For more details on XML, see "Types of input data" on page 240.

To enable the RTF to be formatted correctly, when you import your sample data into a data format you need to ensure that the RTF field is defined as type *string*. While this will be done automatically for XML data, you must define it manually for other data types. "Creating a data format" on page 244 contains information on defining records and fields.

> For more information on RTF fields, see "RTF fields" on page 108. For information on including resources that are required by RTF fields in the HIP file see "Additional Resources" on page 342.



```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE FILE SYSTEM "customerdtd.dtd">
<FILE>
<DOCUMENT>
        <ACCT>Account Number</ACCT>
        <NAME>Customer Name</NAME>
        <ADDRESS1>Customer address line 1</ADDRESS1>
        <ADDRESS2>Customer address line 2</ADDRESS2>
        <ADDRESS3>Customer address line 3</ADDRESS3>
        <TESTIMONIAL>{\rtf1\ansi\ansicpg1252\deff0\deflang1033
        {\fonttbl{\f0\fswiss\fcharset0 Arial;}{\f1\fswiss\fprq2
        \fcharset0 Tahoma;}{\f2\fnil\fcharset2 Symbol;}}
        {\colortbl ;\red0\green0\blue128;}
        {\*\generator Msftedit 5.41.15.1503;}\viewkind4\uc1\pard\f0
        \fs20 An independent survey revealed that:\par
        \pard{\pntext\f2\'B7\tab}{\*\pn\pnlvlblt\pnf2\pnindent0
        {\pntxtb\'B7}}\fi-720\li720\b 92%\b0  of our customers were
        \i 'exceedingly'\i0  or \i 'very'\i0  satisfied with our
        services\par {\pntext\f2\'B7\tab}CSRs could answer most
        customer queries within \b 2 minutes\b0\par \b{\pntext\f2
        \'B7\tab}85%\b0  would recommend their friends switch to
        \cf1\b\f1\fs22 Big Bank\cf0\b0\f0\fs20 .\par
        }</TESTIMONIAL>
</DOCUMENT>
</FILE>
```

An RTF field within an XML input data file.

Localization

DOC1 uses the concept of locales to allow control and customization of data and presentation objects to take account of regional variations. Such requirements may range from simply using regional date and currency formats to complete customization of document content including text translation and conditional resource selection.

About locales

Locales operate on several levels. Locale objects themselves contain default formatting information for most DOC1 data types. For example, you can specify the standard currency symbol, the character used to separate date components, the way in which positive and negative numbers are formatted and so on.

A locale is used as part of a data format to specify the default manner in which date and number values are evaluated when input data is parsed.

ⓘ SEE "CREATING A DATA FORMAT" ON PAGE 244 FOR DETAILS.

The Work Center provides a default locale for publications and the design objects contained within them. By default, English (United States of America) is the locale that is used by publications until you choose to change the setting.

Document designs can reference many locales which are used to determine the default formatting required when data fields, variables and other objects are output.

ⓘ YOU CAN OVERRIDE THESE DEFAULTS ON A CASE-BY-CASE BASIS. SEE "FORMATTING VALUES" ON PAGE 106 FOR DETAILS.

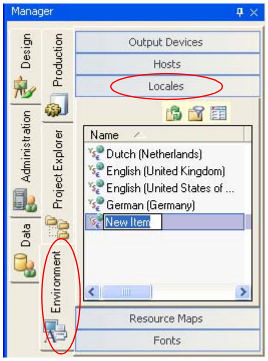
Each design object has a default locale that is active when processing of a document begins. An action in document logic allows you to switch to a different locale at any point during processing.

You can also create localized variations of entire paragraphs, normally so that the text they contain can be translated. The paragraph variants are associated with particular locales and, as with regional settings, are activated when the active locale is switched.

▶ **To set the default locale:** Select the *Environment* tab and then the *Locales* bar. Highlight the locale you want to set as the default and select *Tasks/Set as Default Locale*.

Creating and customizing locales

Locales are stored as independent objects within the DOC1 Work Center and, permissions allowing, can be referenced by any document design or Active Content object. A default set of locale objects is provided with DOC1 distribution material. You can customize these as required and create new locales to suit your needs.



▶ To create a new locale:

1. In the Work Center Manager, select the *Environment* tab and then click the *Locales* bar.
2. From the File menu select *New Locale*.
3. Enter a name for the locale and open it.
4. In the Locale Configuration window specify the settings for each data type as required.

▶ To edit an existing locale: double-click on a locale in the Work Center Manager and edit the settings as required.

Localizing document designs

Each document design has a default locale which sets the default formatting to be used when data fields, variables and other objects are output. The selected locale is automatically activated whenever the document starts to process.

> ⓘ Unless you have changed the default locale for the Work Center, English (United States of America) will be used as the default locale for your design objects until specified otherwise.

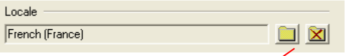
If you are using sections within document designs you can override the locale specified for a document on a section-by-section basis.

At any point within document or publication logic you can specify an action to activate an alternative locale. The new locale will then be used until a further *Change locale* action is encountered or until a new document (or a section with a locale specified) is loaded.

▶ To set the default locale for a document:

1. Double-click the appropriate Document icon in the logic map of the document or publication editor.

2. In the *Document* dialog box use the *Locale* button to select from a list of existing locale objects known to the Work Center


Locale
French (France)

Click to add or remove a locale.

▶ **To override the default locale for a section:** double-click the appropriate Section icon in the document logic map and in the *Section* dialog box use the *Locale* button to select an alternative locale object.

> ⓘ You must ensure that the correct dictionary for the selected locale is used for the spell checker to be accurate.

▶ **To change the active locale:**

1. In the document or publication editor logic map mark the insertion point at the required position.
2. From the Insert menu select *Control objects/Action*.
3. In the *Action* dialog box, select *Change locale* and using the browse button 📁, select a locale to be activated when the action object is encountered.

> ⓘ To ensure that the *Change Locale* action takes effect, you must remove any locales assigned to lower level documents or sections.

Localized paragraphs

You can create variations of paragraph content which are automatically selected according to the active locale. Before you can create localized paragraphs you must specify the locales that are to be available with this feature used (and therefore the number of variants a paragraph can have).

You must indicate the paragraphs that are to be localized and the language (locale) required for each.

Once defined, such paragraphs each have a master version and a variant for each selected language. The content of the master version is automatically copied to all language variants when it is marked for localization (and when new languages are added).

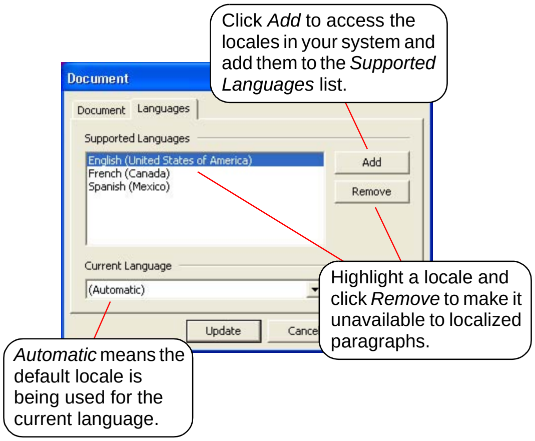
Once content has been applied to language variants it must be maintained independently of the master. However, any new content that is added to the master version is also automatically appended to all language variants from where it can be merged into the localized presentation.

Localized paragraphs can contain in-line or anchored graphics as well as text and these can be maintained independently within the variants.

You can create localized paragraphs within document designs and Active Content objects.

▶ To enable localized paragraphs:

1. For a Document, double-click the appropriate Document icon in the logic map of the document or publication editor.
   – or –
   For Active Content, double-click the Active Content icon in the logic map of the Active Content editor.
2. Select the *Languages* tab.
3. Click the *Add* button and select from the locales (and hence languages) in the system to make them available to the paragraphs in the document.



Click *Add* to access the locales in your system and add them to the *Supported Languages* list.

Highlight a locale and click *Remove* to make it unavailable to localized paragraphs.

Automatic means the default locale is being used for the current language.

▶ To set the available languages for a paragraph:

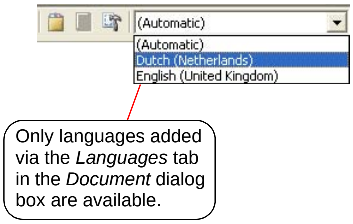
1. Double-click the paragraph icon in the logic map
   – or –
   highlight the paragraph and from the menu select *Format/Paragraph*.
2. In the *Paragraph* dialog box select the *Multiple Languages* check box.

   The languages included in the *Languages* tab of the *Document* dialog box are now available to the paragraph.

Editing localized paragraphs

By default, the document editor will display (and allow you to edit) paragraph variants according to the locale that has been activated by the logic currently in force. For instance, if you have a condition that sets the active locale for one of two languages, the condition that currently evaluates to true will determine the paragraph variants that are displayed.

You can override this default and show all the paragraph language variants for a specific locale if required.



Only languages added via the *Languages* tab in the *Document* dialog box are available.

▶ To work with specific language variants: in the toolbar select the *Language* list and select the localized paragraph language you want to display.

When editing paragraph variants you can do general text editing, such as changing the fonts, line spacing, moving and deleting sections of text. You can also re–arrange field parameters and re–order functions but data fields can only be removed in the master version of the paragraph.

If you split a localized paragraph you will be given an option to create language variants for the new paragraph. However, the language variants of the original paragraph will be unaffected by this and you may need to adjust their content manually to reflect the split.

If you join localized paragraphs the additional text will be applied to all variants of the remaining paragraph using the currently active language. You may need to localize the content of the other language variants as appropriate.

Production journals

DOC1 uses journal files to allow you to record the activity of a publication when it processes in the production environment. You may often want to use journals to make an index for the documents and pages produced by the application. These can help with things such as post-processing requirements, audit trail and archiving functions.

You can write as many entries as required to one or many journal files. During processing in the production environment an entry will be written to the assigned file whenever a journal object is encountered in publication logic. You can use conditions and other logic to determine if and when an entry is actually written. Journals are normally generated within a publication, but you can also generate them at the start and end of the entire production job.

In the output file the entry itself is a new line or record containing the value(s) specified in the journal object. Each value can be separated with a user defined string if required or simply concatenated together.

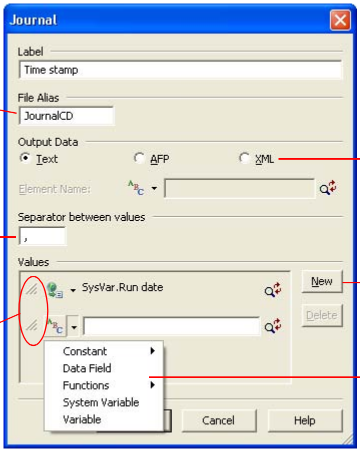
The target output files for journal entries are distinguished by the file alias you specify. Each different alias used in a publication design will produce a separate file in the production environment. You will need to specify the actual file references to receive the journal output when the publication is published. See “Publishing and deployment” on page 329 for details.

DOC1 can create indexes in three formats.

- *Text* – a simple text record is written to the output file each time the journal is processed.
- *AFP* – the index is a structured file in IBM AFPDS format. There are four values per entry.
- *XML* – named elements are written together with their associated values to the journal in XML format.

▶ **To create a journal object**: mark the insertion point in the logic map and select *Insert/Tools Objects/ Journal*. The *Journal* dialog box is displayed.
Specify the journal as required.

Start and end of job journals must be specified in the Publication attributes – see the Journal Options in “Publications and documents” on page 96.



Label: Time stamp

File Alias: JournalCD

Output Data: ● Text ○ AFP ○ XML

Element Name:

Separator between values: ,

Values: SysVar.Run date

New

Delete

This alias associates the entry with an actual file in the production environment. Make sure this matches aliases in any other journal object intended for the same file.

The values specified are output to the journal file in one line or record, separated by this text (up to 8 characters).

Change the order of values by dragging the handles up or down.

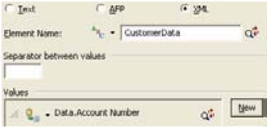
Select the type of journal required.

Create a *New* value …

… and specify it. See “Working with values” on page 104 for details.

Working with XML Journals

When defining XML journal objects both element names and associated values are assigned. Note that the data structure is automatically defined in the Data Type Definition (DTD) embedded in the XML journal itself.



○ Text ○ AFP ● XML

Element Name: CustomerData

Separator between values:

Values: Data.Account Number

AFP Indexing

AFP indexes should only be specified in a DOC1 publication when at least one of the intended output datastream is AFP. The index information is in binary form and is intended for use only with third party software that can deal specifically with this indexing structure.

Format of AFP Indexes

When creating a journal that will be used as an AFP index, each entry must include four values that are used to provide parameters for the AFP index structure. These are:

Attribute The name associated with entries of a particular type.

Attribute value A value that distinguishes this entry from others with the same title.

Sequence number A numeric value that indicates the sequential order of index entries.

Level A numeric value that subdivides index entries with the same attribute title.

The values must be specified in the order shown and you may not omit any value.

In the production environment DOC1 Generate will use journals defined as AFP indexes in two ways:

- The journal file itself will be created as a self-contained AFP index containing all the entries generated by the job.
- For the AFP datastream itself an AFPDS Tag Logical Element (TLE) structured field containing the relevant index values will be included in each page generated by the appropriate publication.

Interfacing with message applications

Message1 and Content Author are optional additions to the DOC1 environment that allow key parts of document content – messages – to be created, maintained and deployed independently of the main publication design. You can think of a message as a segment of page content that can contain text and graphics and make use of variable data available to the publications in which they are used.

Message applications use *message areas* to define the space available to particular groups of messages. In DOC1 Designer you can map such areas to specific reserved space in document designs – *message boxes* – or alternatively message areas can be inserted directly as part of the main flow area of the document – *message streaming*. You will also need map the *data fields* used in messages to specific values provided by your publication logic and, for Message1 only, link the *language* settings used in the messages to the equivalent DOC1 Designer locale.

Information about the message areas and data fields is made available to DOC1 Designer via a *message environment file* which is created by the message application.

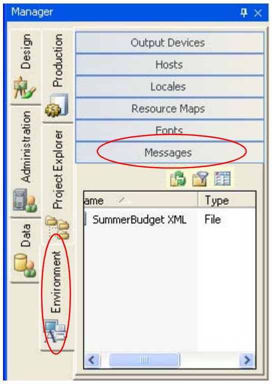
Ideally the administrator of the message creation environment will be the same person as the designer of the DOC1 publications intended to use the messages. Where this is not the case they will need to work closely to agree the available space for messages and to define the values that should be mapped to data fields.

> ⓘ You may only include a single environment file in a DOC1 repository. This is assumed to contain all the data required in all publications that use messages. After the first import subsequent attempts are assumed to provide updates and will overwrite the information for areas and data fields defined in earlier imports.

▶ To import a message environment file:

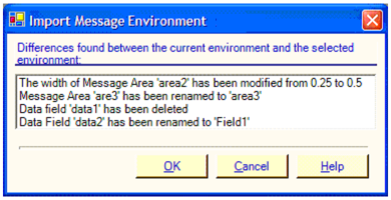
1. In the Work Center Manager select the *Environment* tab and then click the *Messages* bar.

2. From the Tasks menu, select *Import Message Environment*. Use the resulting dialog box to browse for the required environment file (.XML).



3. Confirm updates.

   If you have previously imported an environment file differences with the new file will be indicated. Click *OK* for the information to be updated.



Import Message Environment

Differences found between the current environment and the selected environment:

The width of Message Area 'area2' has been modified from 0.25 to 0.5
Message Area 'are3' has been renamed to 'area3'
Data field 'data1' has been deleted
Data Field 'data2' has been renamed to 'Field1'

▶ **To specify language mappings for Message1:** in the Work Center Manager select the *Environment* tab and then click the *Messages* bar. Use the *Message1 language mappings* tab to set language mappings. You can *add* and *delete* mappings as required. Note that Content Author uses the DOC1 locale for language matching, so it is not necessary to map languages specifically.

Message boxes and areas

A *message box* reserves space in a document design for messages. You must assign one or more *message areas* to each message box to indicate the category of messages it can receive. Such message areas are created in the message application and specify the maximum dimensions of message content. The message box must always be large enough to accommodate the areas that have been assigned to it.

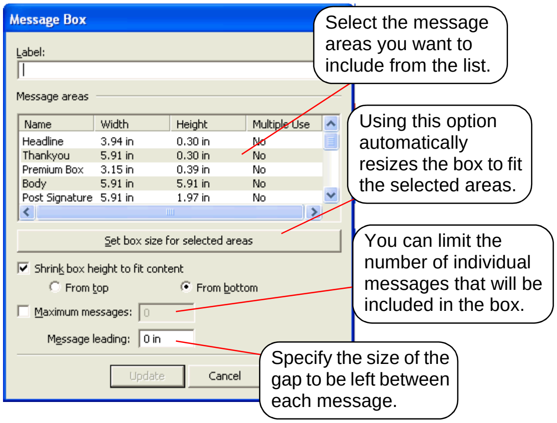
ⓘ Some types of message area allow individual messages to be repeated within the same document (*multiple use*). You cannot mix such areas with non-repeating areas within the same message box.

Message boxes behave similarly to text boxes. They can be inserted anywhere in a document design, for instance, in the body area, on the background, in headers and footers. Their attributes can be adjusted as with any other shape object. See “Working with graphics” on page 48 for details.

By default the width and the maximum height of a message box is fixed by the dimensions you specify. If required, you can use the *Shrink box height to fit content* options to specify that the height of the box should vary where the content is less than the specified height.

▶ To create a message box:

1. Mark the required insertion point within the logic map.
2. From the *Insert/Presentation Objects* menu select *Message Box*
3. Create the box. by clicking in the document design and drag out the box to approximately the required size. This can be adjusted later.
4. Specify message areas and customize. Double-click the message box object to show its properties.



Message Box

Label:

Message areas

Name	Width	Height	Multiple Use
Headline	3.94 in	0.30 in	No
Thankyou	5.91 in	0.30 in	No
Premium Box	3.15 in	0.39 in	No
Body	5.91 in	5.91 in	No
Post Signature	5.91 in	1.97 in	No

☑ Shrink box height to fit content

○ From top ● From bottom

☐ Maximum messages: 0

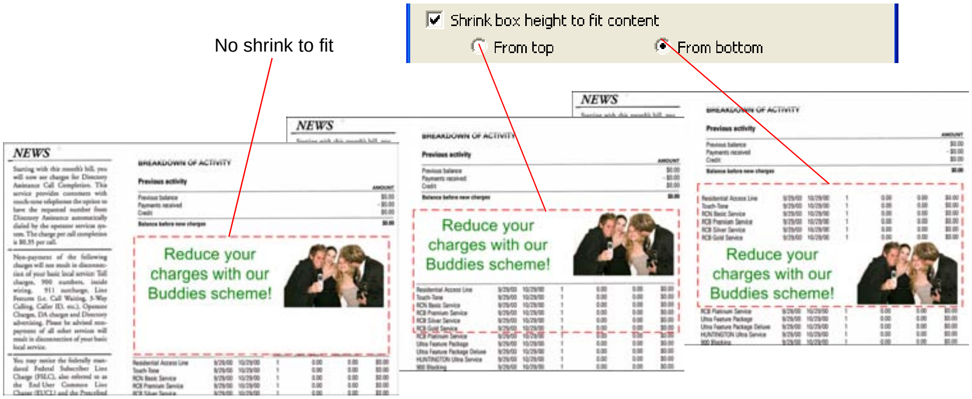
Message leading: 0 in

Select the message areas you want to include from the list.

Using this option automatically resizes the box to fit the selected areas.

You can limit the number of individual messages that will be included in the box.

Specify the size of the gap to be left between each message.



No shrink to fit
☑ Shrink box height to fit content
○ From top
● From bottom
NEWS
BREAKDOWN OF ACTIVITY
Reduce your charges with our Buddies scheme!

Message Streaming

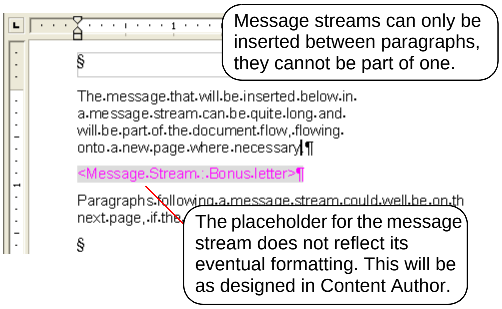
Message streams allow messages that have been created in Content Author to be inserted as part of the main flow area in a document. The content of a message area is inserted as a continuous, flowable unit that will overflow onto new pages as necessary.

Formatting of the messages is as defined within Content Author, with the following exceptions:

- the width is that of the flow area, subject to any indents specified in Content Author
- there is no restriction on the height – the message stream will flow over as many pages as required.

Although the size of the message area is not relevant, the criteria for including an area is used.

Message streams can only be part of the main flow area; they cannot be inserted on the background, in Active Content or page setup, or in a text box or table.



§

The.message.that.will.be.inserted.below.in.
a.message.stream.can.be.quite.long.and.
will.be.part.of.the.document.flow,.flowing.
onto.a.new.page.where.necessary.¶

<Message.Stream.:.Bonus.letter>¶

Paragraphs.following.a.message.stream.could.well.be.on.th
next.page,.if.the.

§

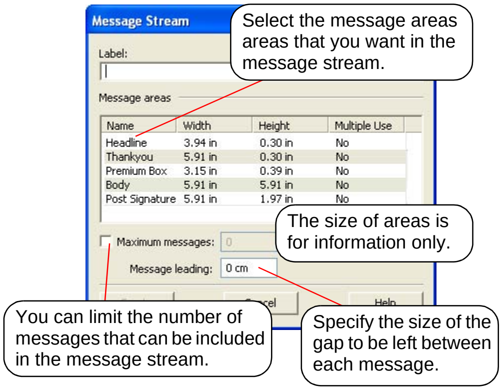
Message streams can only be inserted between paragraphs, they cannot be part of one.

The placeholder for the message stream does not reflect its eventual formatting. This will be as designed in Content Author.

▶ **To insert a message stream:** from the *Insert* menu select *Presentation Objects/Message Stream* and select the required message areas.

As with message boxes, you must map message data fields before messages are actually placed in the message stream. See “Resolving message data fields” below.

Note that if the cursor is at the start of a paragraph, then the message stream will be inserted before that paragraph, otherwise it will be inserted after.



Label:

Message areas

Name	Width	Height	Multiple Use
Headline	3.94 in	0.30 in	No
Thankyou	5.91 in	0.30 in	No
Premium Box	3.15 in	0.39 in	No
Body	5.91 in	5.91 in	No
Post Signature	5.91 in	1.97 in	No

□ Maximum messages: 0

Message leading: 0 cm

Select the message areas areas that you want in the message stream.

The size of areas is for information only.

You can limit the number of messages that can be included in the message stream.

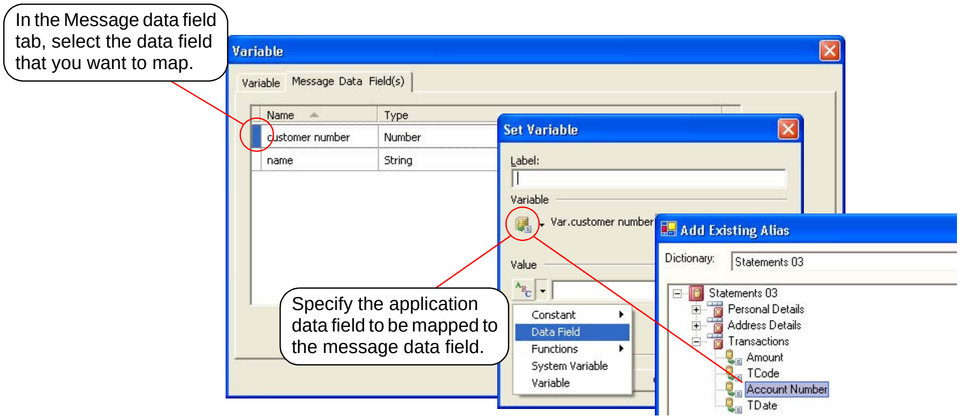
Specify the size of the gap to be left between each message.

Resolving message data fields

See also the “Publish Wizard options“ for “Messages” on page 345.

Messages often include references to values available to a publication at production run-time; these are known as *message data fields*. In a publication design you must ensure that each data field likely to be used in messages is mapped to the required value using the *Set variable* action. In your publication logic you should ensure such mappings are defined before any messages are output but you can update them at any point in the logic (within a condition for example. Once the mapping is defined it will stay in force until a further action specifying the same data field is encountered.

▶ **To map a message data field:** mark the insertion point in your document logic and select *Insert/Set variable* from the menu.



In the Message data field tab, select the data field that you want to map.

Specify the application data field to be mapped to the message data field.

Production & preview

You can preview how the messages will look in your publication by using the advanced preview feature, see “Previewing a publication” on page 272. You can also preview messages from within the message application. You will need to generate a .HIP file (this must be a complete publication containing both design and resources) for the intended DOC1 publication in the normal way ensuring that the output file *Output1* is set to *PDF*. Make both the .HIP file and the sample data file available to the message application. Note that for Message1 no other output devices should be specified.

When publishing and previewing, you will need to specify the Publish wizard options that are required for message applications including the file containing the latest messages and error handling options – see the “Publish Wizard options“ for “Messages” on page 345.

Two system variables – *Mandatory messages* and *Optional messages* – allow you to monitor the status of messages that are available for placement within the document currently being processed and optionally activate appropriate logic. See “System variables” on page 155 for details.

Customizing the audit trail

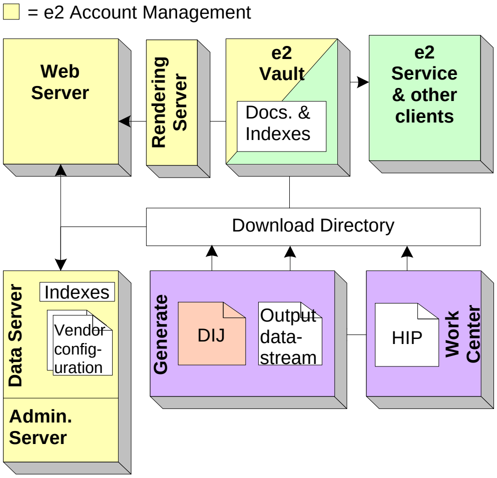
For DOC1 publications that use messages Generate creates an additional log file known as the Message Audit Trail. This is similar to a regular DOC1 journal file but is specifically and automatically used to log messages that have been included within documents. In the Designer you can assign a field that will be additionally included in the audit trial entries along with the automatically generated references to messages. This is intended to allow the user to generate a reference that can be used in the audit trail to identify documents uniquely – typically a field containing a customer reference or similar. The *Set message audit trail* option can be re-specified at different points in the logic if required.

▶ **To specify the unique reference for the audit trail:** mark the insertion point in your document logic, (ensuring that this is before any messages are placed by the logic – i.e. before the first message box) and select *Insert/Control Objects/Action* from the menu. In the *Action* dialog box select the *Set customer ID* option and assign the required field or value.

Interfacing with e2 Vault

The e2 Vault environment caters for high speed storage and retrieval of electronic documents. e2 Vault itself is the main repository in which documents are stored and maintained. Various other e2 products provide access to the repository.

e2 Account Management is an extension to the core Vault environment that allow web based display of billing data to end users, optionally with payment facilities. While it accesses the Vault repository, e2 Account Management has its own management systems.



e2 Vault can accept documents from several sources and has a variety of methods for loading the data and associated resources. For documents created by DOC1 applications Vault always requires the following resources:

- The actual documents to be archived as part of a standard output datastream file created by DOC1 Generate
- A DOC1 Interchange Journal (DIJ) to act as an index for the documents also created by Generate
- The resources required to present the documents delivered within the appropriate HIP file (as created by a Work Center publishing task).

For information about loading these files into e2 Vault refer to the e2 Vault Users Guide.

If the archived documents are also to be used within a e2 Account Management environment the application also needs to provide vendor configuration data which is stored within the DIJ. Normally the data required can be created as part of the relevant publication object within the DOC1 Work Center.

Creating the index

The e2 Vault repository has a very fast access method that relies on a well constructed index into the documents stored in the datastreams it holds. For documents generated by DOC1 Generate this index is provided by a DOC1 Interchange Journal (DIJ).

Each record within a DIJ specifies a unique key for each document by which it will be referenced in the Vault environment (usually an account number). A DIJ can also provide additional information about the customer account to which a document belongs, to provide search data for use with Vault client systems.

Custom presentation

If required, you can also write publication data to the DIJ for custom presentation by e2 Present. When you use the *Store Data Set* option, the input data destined for each document is copied to the DIJ along with the regular index information you have requested. Contact your DOC1 supplier for further information if you want to use this option.

DIJ settings for use with e2_Vault

This list shows how the information in a DIJ can be used by e2 Vault.

> See “Appendix C – DOC1 Interchange Journals” on page 419 for more details.

Data tab

these parameters are used to identify individual publications within the Vault repository. Together they provide the primary index key within the repository

Account Number – a unique reference number for the intended recipient of the publication (usually an account number).

Name – a name that uniquely identifies the application generating the publication.

Statement Date – a date that uniquely identifies the version of the publication

Publication ID tab

use this section for unique identification of the company (vendor) and type of publication to which documents relate. Under normal circumstances the *Default* setting, using generic settings, will provide the information required.

Master ID – you may need to specify this if your e2 Present implementation uses a custom payment type.

Read from .PSE file – you can use a .PSE file generated by e2 Present that contains the relevant ID settings.

Options tab

City, State/Region, Zip/Post Code, Country & *Phone* – these provide additional search options within Vault client systems.

Payable – select this option if the documents are to be used with e2 Present.

Address tab — use this to provide the address data related to the account to which each document belongs.

Custom settings tab — use for additional document index information that may be required where custom versions of e2 Present are in use.

Table of contents

For more information about creating bookmarks see “Bookmarks” on page 93.

For extended document types you may want to include a **table of contents** that is available when documents are viewed from the Repository. Where used the structure and labels used in the TOC are specified by using Bookmark objects in the logic map of document designs (i.e. they are not configured in the *DIJ* dialog box. At production run-time, the TOC information will be automatically generated and stored in the DIJ if bookmarks have been used. Bookmarks can be nested to create the required structure for the table of contents.

Interfacing with e-Messaging

The e-Messaging system manages e-mail and SMS (Short Message Service) customer communications. It allows organizations to design content for both these in a single design environment. It enables organizations to allow their customers to communicate at any time, maintaining an up-to-date customer-centric view at all times throughout the organization. It can also help organizations to maintain a brand consistency

> See “Appendix C – DOC1 Interchange Journals” on page 419 for more details.

The DOC1 Document Interchange Journal (DIJ) is used as the main control file and contains information, such as customer account numbers, which references the corresponding publications in the DOC1 output datastream via an e-Messaging Identifier. The following output datastreams are available for e-Messaging:

eHTML for customers wishing to view their information as rich text.

Line data for customers who prefer to view their information in plain text and on devices that can only support plain text. This driver can also output other text ready for insertion into other streams and web pages by e-Messaging.

DIJ settings for use for use with e-Messaging

This list shows what information in a DIJ can be used by e-Messaging.

> Refer to the e-Messaging Reference Guide for details.

Data tab use all fields to identify the documents to be sent in the messages. *Statement Date* should be the date the message is to be sent, as it will appear in the e-mail header.

Publication ID tab not required – use the *Default* setting.

Options tab not required, but can be used for extra address fields.

Address tab Address fields can be included as required.

Custom settings tab Contact your DOC1 supplier for information on these settings.

Creating EngageOne templates

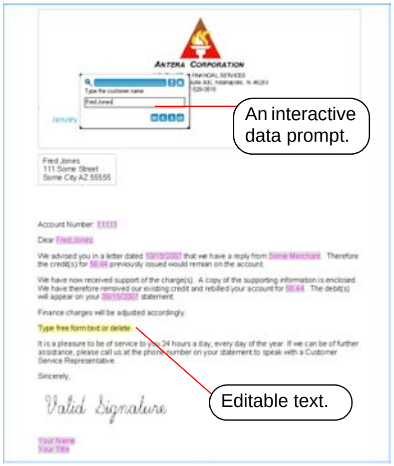
> THIS SECTION COVERS THE CONSIDERATIONS THAT SHOULD BE GIVEN TO PUBLICATION DESIGN WHEN CREATING ENGAGEONE APPLICATIONS. FOR COMPLETE INFORMATION ABOUT ENGAGEONE PLEASE REFER TO THE ENGAGEONE ADMINISTRATION GUIDE.

EngageOne is a solution that allows the creation, control and delivery of ad hoc customer correspondence such as letters and offers based on a standard DOC1 publication design. In this scenario the DOC1 Designer creates a **template** which will be launched and interactively customized by a front-office user (known as the **correspondent**) before printing the document or submitting it for further processing.

The layout and style of an ad hoc document may be very different from the designs used for batch applications but within DOC1 Designer the files and objects used are fundamentally the same. It is only at publishing time that a publication is identified as an EngageOne resource and results in a template file being created for use with the EngageOne environment.

At the design level the primary difference between a regular publication and one intended for EngageOne is the use of an **interactive data definition.** This replaces the data format used to define input data for regular publications. The interactive data definition defines each piece of variable data required by an EngageOne application and specifies whether the value should be provided directly by the correspondent (prompted) or automatically from a linked external system. The dictionary and data map is created from this interactive data and linked to the template when it is created.

Where more flexible content is required you may also specify that the correspondent is able to directly amend parts of paragraph text. To the correspondent this is known as **editable text**.



Fred Jones
111 Some Street
Some City AZ 55555

Account Number: 11111

Dear Fred Jones

We advised you in a letter dated 10/15/2007 that we have a reply from Some Merchant. Therefore the credit(s) for $0.44 previously issued would remain on the account.

We have now received support of the charge(s). A copy of the supporting information is enclosed. We have therefore removed our existing credit and rebilled your account for $0.44. The debit(s) will appear on your 09/15/2007 statement.

Finance charges will be adjusted accordingly.

Type free form text or delete.

It is a pleasure to be of service to you 24 hours a day, every day of the year. If we can be of further assistance, please call us at the phone number on your statement to speak with a Customer Service Representative.

Sincerely,

Valid Signature

Your Name
Your Title

Active Content objects also play an extended role in EngageOne applications. A typical EngageOne template will make use of one or more Active Content groups. Such groups can be used, for example, to automatically select a range of paragraphs when particular properties are matched and provide a list of possible paragraphs to be included in the final document.

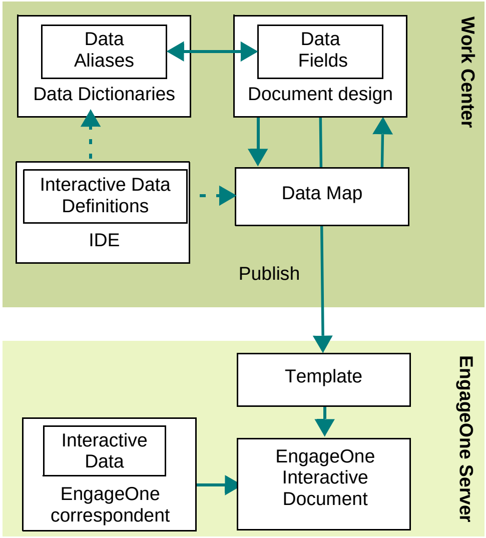
Using Active Content objects can also significantly reduce the time and resources needed to deploy content changes to the live environment as they can be published and applied separately from the main template files. Additionally, organizations may often want to maintain many different EngageOne templates with similar content so using shared Active Content can allow changes to be easily distributed. For full details of working with Active Content see “Using Active Content” on page 224.

In the EngageOne environment, other actions may occur within the document based upon the action taken. A new list of choices may become available, a new piece of content may automatically be added as an attachment, or conversely, content and choices may be removed based upon a user’s choice.

If you publish for EngageOne, templates do not contain any font and image resources for target output datastreams. Such resources are placed within independent files that must be created and deployed in conjunction with requests from the EngageOne server. This architecture allows EngageOne to re-use previously deployed resources wherever possible. For more information see “Publishing resources for EngageOne” on page 218.

Before you deploy new or amended EngageOne resources to the server a *Preview for EngageOne* function allows you to see a template as it will appear to the correspondent; i.e. loaded in a web browser and with interactive data prompts displayed.

EngageOne data



An **interactive data definition** describes all the interactions required to complete a document from the template in the EngageOne environment. They are created and edited in the **interactive data editor** (IDE). Once the definitions are saved, the interactive data format is created and the **data map** and **data dictionary** are generated. For more information about data maps and data dictionaries see "About DOC1 input data" on page 237.

Note that data definitions are typically created before you build the EngageOne template. The interactive data can then be referenced and incorporated into the template design by assigning the data map that was generated from the IDE.

The *Preview for EngageOne* feature allows you to view the prompts you created using the IDE as they would be seen by a correspondent.

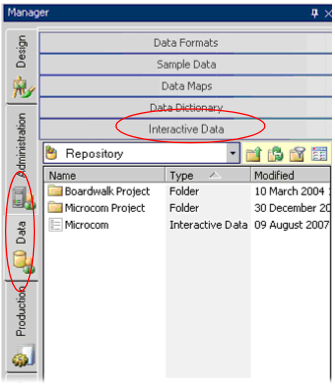
Using the Interactive Data Editor

The interactive data editor is used to define all data that is required for the EngageOne document, including:

- data gathered from the correspondent via on-screen prompts
- data supplied directly from external sources such as system supplied data, to be used directly in the publication.

When you define fields these can be included in *groups* containing several fields of hierarchical data for defining prompted fields, or *repeating groups* of system supplied data.You can specify default values for fields within repeating groups for an unlimited number of repetitions of the group.

You can also use the IDE to define annotations for any data field provided they comply with the W3C XML schema annotations standard. Annotations allow you to pass additional information through the EngageOne schema to your system data integration program. For more information about integrating external system data see the EngageOne Programmer's Reference Guide.

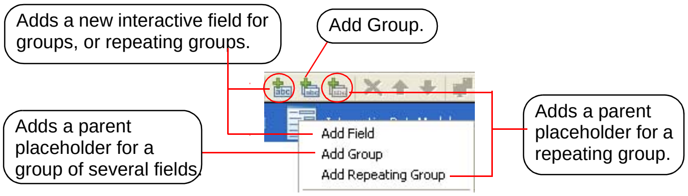


▶ To create an interactive data definition:

1. In the Work Center Manager select the *Data* tab and then click the *Interactive Data* bar.
2. From the *File* menu select *New Interactive Data*.
   – or –
   right-click in the publication explorer pane and select the *New Interactive Data* option.
3. The main window of the IDE is invoked.

▶ To add fields, groups and repeating groups:

1. Use the add buttons on the toolbar
   – or –
   right-click on the *Interactive Data Model* button to display the shortcut menu. Add fields as required.



Adds a new interactive field for groups, or repeating groups.

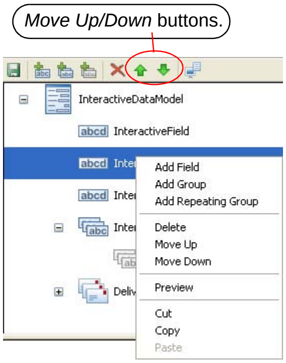
Add Group.

Adds a parent placeholder for a group of several fields.

Adds a parent placeholder for a repeating group.

2. Use the *Field Properties* pane to define the properties and values of the currently selected field. For details see “ Field properties – Interactive Data Editor” on page 205.
3. Click *OK* after completing each field definition.

▶ To re-order fields, groups and repeating groups:

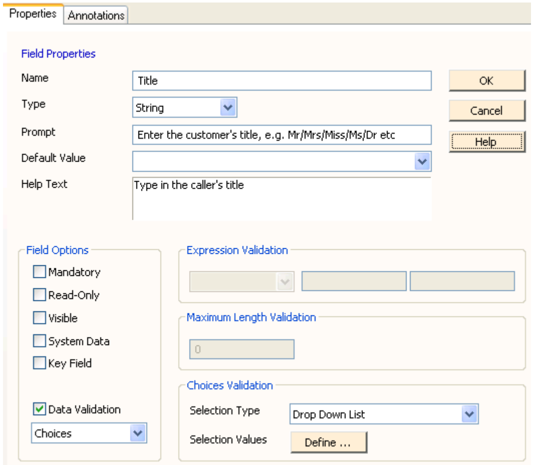

Move Up/Down buttons.

1. Adjust the field list as required by using the toolbar
   – or –
   use the shortcut menu by right-clicking on any *Field*, *Group* or *Repeating Group*.
2. Options include: *Cut* and *Paste* to move groups and fields as required on the list and *Copy* and *Paste* to copy items and add them either to a position within the same level, or to a different level or group. A copied item will take the original name plus an appended number. Rename this as required in the *Field Properties* pane. Use the *Move Up* and *Move Down* buttons, or the shortcut menu to move the selected field up and down the list view (within the same level in the tree). You can also drag and drop fields to move items one position up and down the tree.

   Note that the *Delivery Information* fields are predefined and can not be renamed or deleted, and the *Type* can not be changed.

3. Use the *Properties* tab to define the properties and values of the currently selected field from the field list.
4. Once all interactive fields have been defined click the *Save* button on the toolbar. This saves the IDE and issues the interactive data definition.

Field properties – Interactive Data Editor



Field Properties

Name: Title

Type: String

Prompt: Enter the customer's title, e.g. Mr/Mrs/Miss/Ms/Dr etc

Default Value:

Help Text: Type in the caller's title

Field Options

☐ Mandatory

☐ Read-Only

☐ Visible

☐ System Data

☐ Key Field

☑ Data Validation

Choices

Expression Validation

Maximum Length Validation: 0

Choices Validation

Selection Type: Drop Down List

Selection Values: Define ...

Name A unique name to identify the field or group item. Must not contain spaces.

Type Identifies the type of data the field contains: Select one of the following from the drop-down list.
Integer – defines a whole number. Type the number in the *Default Value* field or select the number using the drop-down list.
Number – defines a decimal number. Type the number in the *Default Value* field.
String – defines a text field. Type the text in the *Default Value* field.
Date – defines a date. Click the drop down arrow in the *Default Value* field to display a calendar control, localized to the regional setting of the local computer. Select the date required.
Time – defines a time of day value. Click the up and down arrows to change the time or enter the value directly in hours, minutes and seconds.

Prompt Optionally defines the message to be displayed to describe the action required from the user during data capture.

Default Value Optionally represents the initial value that the interactive field will contain when the template is first loaded.

Help Text Optionally defines an extended message for the *Help* button to provide more comprehensive instructions to the user.

(continued)

Field properties – Interactive Data Editor

Mandatory — Marks the interactive field as compulsory for completion. When filling out the template a value must be provided for this field in order to progress to the next interactive field or to generate the document.

Read-Only — Specifies whether field values are user-changeable or not. Fields marked as read-only are not prompted for entry.

ⓘ THIS PROPERTY WILL BE FREQUENTLY SET ON FIELDS CONTAINING SYSTEM-SUPPLIED DATA FROM MAINLINE BUSINESS SYSTEMS.

Visible — This is an advanced option that dictates whether the field will appear in the view when the form is displayed in an XForms compatible web browser. For more information about XForms and EngageOne data integration techniques see the EngageOne Programmer's Reference Guide.

System Data — Identifies fields where values are supplied by the mainline business system.

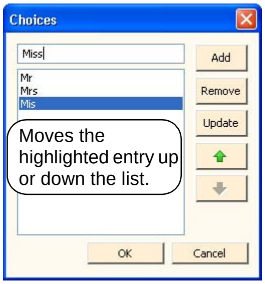
Key Field — Indicates to mainline business systems that this is a primary, searchable field. Also used in the EngageOne Interactive Correspondent application to identify records in the user's *Work in Progress*.

Data Validation — Select *Choices*, *Expression* or *Max Length* from the drop-down list to open the appropriate fields required to define the values. Available options depend on the *Type* of field selected.

Expression Validation — Fields are opened to define data fields if *Data Validation/Expression* was selected on non-numerical field *Type*. Select one of the following from the drop-down list.
Range of – opens the next two adjacent fields for entering the range values. The "from-to" values can be entered, or selected from the drop-down list.
Greater than – opens the adjacent field. Type the value or select one from the drop-down list.
Less than – opens the adjacent field. Type the value or select one from the drop-down list.

Maximum Length validation — This field is opened if *Data Validation/Max Length* was selected. Specify the number of characters allowed for a *String Type* field.

Choices Validation — These fields are opened to define data fields if *Data Validation/Choices* was selected in the numerical field *Type*. In the *Selection Type* drop-down list define how a **single** selection from a pre-defined list should be presented to the correspondent, by selecting:
Drop Down List or
Radio Buttons.You can define a list of *Selection Values* that will be made available to the correspondent by clicking the *Define* button to display the *Choices* dialog. One of these values can be selected by the correspondent during data capture.



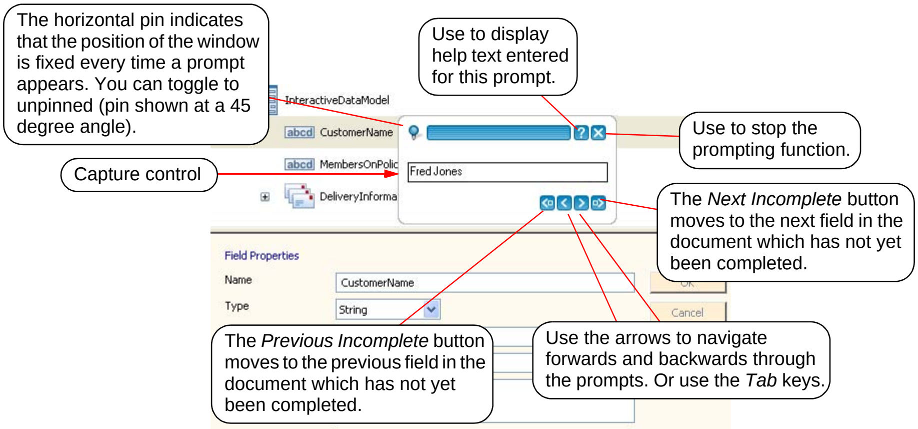
Moves the highlighted entry up or down the list.

(continued)	**Field properties – Interactive Data Editor**
	Browse a predefined list of choices to be made available to the correspondent as possible field values. Add these values using the + button from the *Choices* dialog box. One of these values can be selected by the correspondent during data capture.

▶ **To edit IDE definitions:** click on the field in the list you want to edit and change as required. See “ Field properties – Interactive Data Editor” on page 205 for details.

▶ **To delete IDE definitions:** click on the field or group in the list you want to delete. Click the Delete ✕ button on the toolbar, or use the shortcut menu. Note that if you delete a group all definitions will be deleted with it.

▶ **To preview IDE definitions:** use the *Preview* button in the toolbar to display how your interactive data prompts will be seen by the correspondent. You can move through the prompts using the capture control,
– or –
double-click on the field.



Fred Jones

Name: CustomerName

Type: String

The horizontal pin indicates that the position of the window is fixed every time a prompt appears. You can toggle to unpinned (pin shown at a 45 degree angle).

Capture control

Use to display help text entered for this prompt.

Use to stop the prompting function.

The *Next Incomplete* button moves to the next field in the document which has not yet been completed.

The *Previous Incomplete* button moves to the previous field in the document which has not yet been completed.

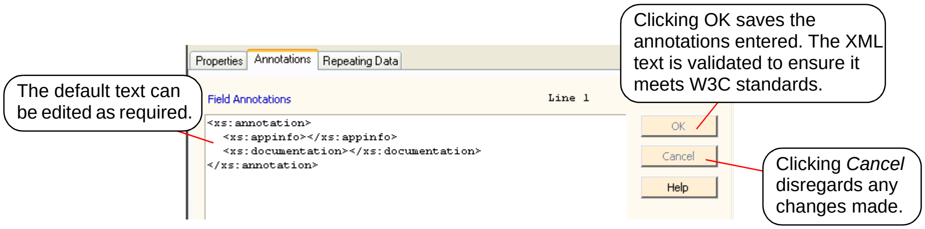
Use the arrows to navigate forwards and backwards through the prompts. Or use the *Tab* keys.

Using annotations

Use the annotations feature to create a self-documenting schema. You are not limited to a line length in Designer, only by the limits imposed by the XML standard. Annotations are definable on groups, fields, repeating groups and delivery nodes.

▶ To add, or edit annotations:

1. Click on the item in the tree that you want to create annotations for.
2. Click the *Annotations* tab to display the *Field Annotations* pane.



```
<xs:annotation>
  <xs:appinfo></xs:appinfo>
  <xs:documentation></xs:documentation>
</xs:annotation>
```

The default text can be edited as required.

Clicking OK saves the annotations entered. The XML text is validated to ensure it meets W3C standards.

Clicking *Cancel* disregards any changes made.

3. Click the *Save* button on the toolbar to save the data definition.

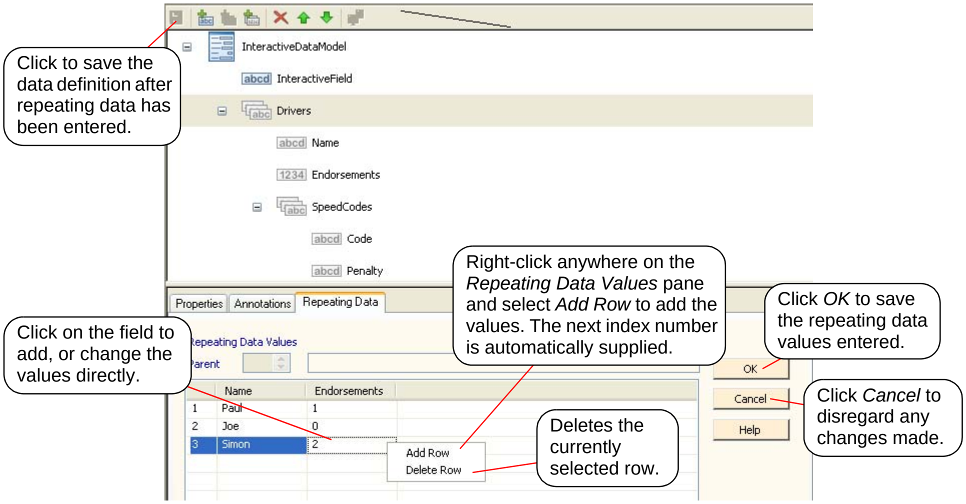
Setting default data for repeating groups

If you have defined and selected a repeating group, a tab is enabled to allow you to add default values for each piece of repeating data within the repeating group. When the data definition is saved, the instance file containing the specified default values is automatically generated in the instance XML.

To present repeating values in your publication use either the repeating data feature – see “Using a repeating data control” on page 179, or set up a transaction table using the Control Data page to identify the repeating data element that will be displayed – see “Repeating data and transaction tables” on page 74. In either case use the data assigned to the publication – see “Generating a dictionary and data map for IDE” on page 210.

Be aware that if you try to change the current data type of a repeating group field with existing default data values, a warning message box is issued. If you choose to accept the change, any existing values will be removed.

▶ **To set default data for repeating groups:** click on the repeating group for which you want to set default data and click the *Repeating Data* tab to display the *Repeating Data Values* pane.



	Name	Endorsements
1	Paul	1
2	Joe	0
3	Simon	2

Click to save the data definition after repeating data has been entered.

Click on the field to add, or change the values directly.

Right-click anywhere on the *Repeating Data Values* pane and select *Add Row* to add the values. The next index number is automatically supplied.

Click *OK* to save the repeating data values entered.

Click *Cancel* to disregard any changes made.

Deletes the currently selected row.

In the case of a nested, repeating group the entered values are assigned a read only, sequential index number. The values are matched to the corresponding index number of the parent node of repeating data. Note that if you set more nested, repeating data values than there are repeating data nodes, a nil record will be set up for those not matched.

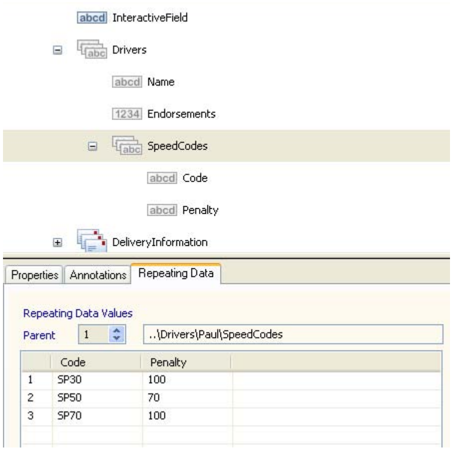
See the following XML example.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<InteractiveDataModel version="1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="b271a7604aab4c4090686e40ad30daf5.xsd">

    <Publication>
        <InteractiveField xsi:nil="true"/>
        <Drivers>
            <Name>Paul</Name>
            <Endorsements>1</Endorsements>
            <SpeedCodes>
                <Code>SP30</Code>
                <Penalty>100</Penalty>
            </SpeedCodes>
        </Drivers>
        <Drivers>
            <Name>Joe</Name>
            <Endorsements>0</Endorsements>
            <SpeedCodes>
                <Code>SP50</Code>
                <Penalty>70</Penalty>
            </SpeedCodes>
        </Drivers>
        <Drivers>
            <Name xsi:nil="true"/>
            <Endorsements xsi:nil="true"/>
            <SpeedCodes>
                <Code>SP70</Code>
                <Penalty>200</Penalty>
            </SpeedCodes>
        </Drivers>
        <DeliveryInformation>
            <Recipient xsi:nil="true"/>
            <EmailToAddress xsi:nil="true"/>
            <EmailFromAddress xsi:nil="true"/>
            <EmailSubject xsi:nil="true"/>
            <EmailBody xsi:nil="true"/>
            <FaxNumber xsi:nil="true"/>
        </DeliveryInformation>
        <g1private/>
    </Publication>
</InteractiveDataModel>
```

▶ To set default data for a nested repeating group:

1. Click on the nested repeating group for which you want to set default data.
2. Click the *Repeating Data* tab to display the *Repeating Data Values* pane.
3. Select the *Parent* node that you want to add values to using the up and down arrows.
4. Add a row by right-clicking anywhere on the pane and selecting *Add Row.* The next sequential index number is automatically supplied.
5. Enter the values as required.
6. Click *OK* to save the repeating data values.
7. Click the *Save* button on the toolbar to save the data definition.



Parent 1 ..\Drivers\Paul\SpeedCodes

	Code	Penalty
1	SP30	100
2	SP50	70
3	SP70	100

Generating a dictionary and data map for IDE

The *Generate Dictionary* option automatically creates a dictionary and data map from the interactive data format. The data elements in the data format are replicated as records and fields in the new dictionary with the mappings already completed in the new data map. See

"Generating a dictionary and data map" on page 262 for more detailed information. When you publish the template for EngageOne you can then assign the data map created to the template publish – see "Publishing for EngageOne" on page 218.

▶ **To generate a data dictionary and data map:**

1. Select the *Data* tab and then click the *Interactive Data* bar.
2. Highlight the interactive data format that you want to use for the data map from the navigator.
3. Select *Tasks/Generate Dictionary* to display the Data Dictionary and Data Map Wizard. The wizard will guide you through the generation process.

   This same wizard can be used to update the dictionary when new fields have been added to the interactive data definition.

Building the EngageOne template

After defining the interactive fields applicable to your template design using the interactive data editor, you can build templates for your EngageOne document as you would a regular DOC1 publication. Interactive fields are referenced as you would other data fields within DOC1 Designer.

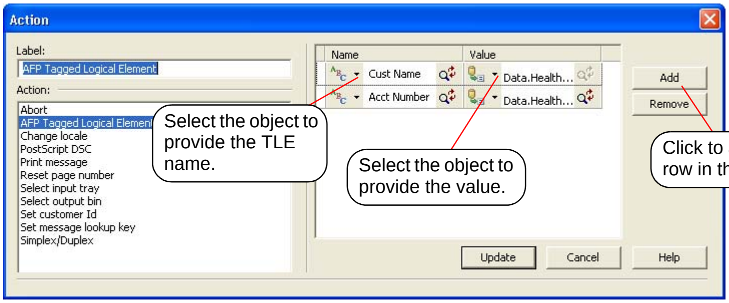
Note however, that the following features are not supported when working with EngageOne templates in DOC1:

Interactive data inserted in the main header or footer is not prompted for, but if referenced in the main body of the document will be simultaneously updated.

- The message box feature (Content Author and Message1) as well as lookup tables are not appropriate in the EngageOne environment.
- Document Interchange Journal (DIJ) and journal entries are not supported through DOC1 Designer for EngageOne documents but can be created using the EngageOne Administration Web application. See your EngageOne Administrator for details.
- The ACIF journal mechanism, Tag Logical Element (TLE) is not supported on the EngageOne platform. If you want indexed output for AFP you can specify the relevant index values at the appropriate point in the logic map.

▶ To create AFP index values for EngageOne:

1. From the *Insert* menu select *Control Objects/Action*.
2. In the *Action* dialog box select an action type of *AFP Tagged Logical Element*.



Action

Label: AFP Tagged Logical Element

Action:

Abort
AFP Tagged Logical Element
Change locale
PostScript DSC
Print message
Reset page number
Select input tray
Select output bin
Set customer Id
Set message lookup key
Simplex/Duplex

Name	Value
Cust Name	Data.Health...
Acct Number	Data.Health...

Select the object to provide the TLE name.

Select the object to provide the value.

Click to add new row in the list.

Active Content and EngageOne

Unlike other uses of Active Content when working with groups you may want to create Active Content objects exclusively for use by a single document design. *Private* Active Content is created and maintained within a document and is not visible outside that document.

Active Content plays an important role in defining optional content in EngageOne documents. Active Content objects are used within Active Content groups, which when marked as interactive allow content to be built dynamically according to selections made by the correspondent from the optional content list.

In a typical EngageOne scenario, you could create an optional content list, with a variable set in one of the entries in the group. Should this variable be matched in the selection you could direct the correspondent to another interactive list elsewhere in the document. Additionally you could automatically add other content later in the document according to a selection made by setting the *Show With* property in a non-interactive Active Content group.

For more detailed information about using Active Content in this way see “Active Content groups” on page 230.

Editable Text

You can allow the correspondent to edit or create free form text as required within an EngageOne document. Such editable text is defined in an **editable container** object in DOC1 Designer and can be limited to a specific string within a paragraph or sentence, or you can allow an entire paragraph to be editable. An editable container can contain editable text (highlighted in yellow), as well as protected, non-editable text. Empty editable paragraphs that are created within a container do not appear in the print stream.

You can create an editable container and insert or mark text as editable within it. Or you can drag the mouse over the required area of text or blank space in the document and mark that as editable, which will also create an editable container. You are also able to drag paragraph objects into and out of editable containers as well as between containers.

ⓘ YOU CAN ONLY MARK ONE PARAGRAPH OF TEXT AT A TIME AS EDITABLE.

In preview mode, the editor toolbar provides standard text editing and formatting features and additional EngageOne related features. You can insert text into an editable container from an external document using copy and paste and use the standard text editing features, bullets and numbering, as well as paragraph formatting (on highlighted text only). For more details see “Previewing an EngageOne template” on page 216. The correspondent will be provided with the same kind of editing features in their front-office application.

ⓘ NOTE THAT IMAGES AND TABLES CAN NOT BE INSERTED INTO EDITABLE CONTENT.

Note that you can only select fonts in the preview that have actually been used within the template design, but when you publish for *Interactive* you can specify extra fonts for the correspondent to use with editable text – see “Running a publishing task” on page 336.

▶ To create an editable container:

1. In the Work Center Manager select the *Design* tab and then click the *Publications* bar.
2. Select the required template.

3. Select the position in the document, or in logic map where you want to place an editable container.

4. From the *Format* menu select *Mark as Editable*.
   – or –
   select the *Mark as Editable* button on the *Standard* toolbar.

5. The *Editable Container* dialog box is displayed.

6. Assign an identification label and click *Create*.

7. Enter or copy text into the container as required. The text will automatically be marked as editable. To revert editable text back to protected text see page 215.

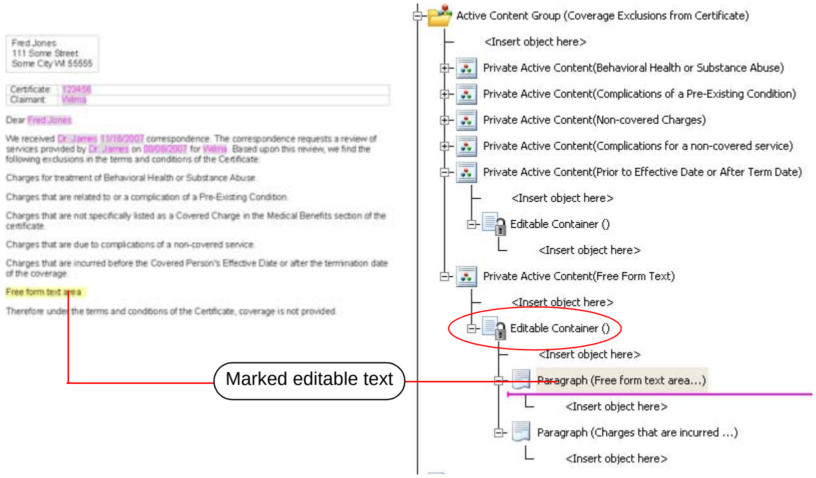
▶ To mark text as editable or protected:

1. In the Work Center Manager select the *Design* tab and then click the *Publications* bar.

2. Select the required template.

3. Drag the mouse over the required area of text or blank space in the document that you want to mark as editable and from the *Format* menu select *Mark as Editable*.
   – or –
   select the *Mark as Editable* button on the *Standard* toolbar.
   – or –
   use CTRL+E.

> NOTE THAT IF YOU MARK TEXT AS EDITABLE IN A DOCUMENT WITHOUT CREATING THE EDITABLE CONTAINER FIRST, THEN THE EDITABLE CONTAINER WILL AUTOMATICALLY BE CREATED FOR YOU.



Fred Jones
111 Some Street
Some City WI 55555

Certificate	123456
Claimant	Wilma

Dear Fred Jones

We received Dr. James 11/16/2007 correspondence. The correspondence requests a review of services provided by Dr. James on 08/08/2007 for Wilma. Based upon this review, we find the following exclusions in the terms and conditions of the Certificate

Charges for treatment of Behavioral Health or Substance Abuse.

Charges that are related to or a complication of a Pre-Existing Condition.

Charges that are not specifically listed as a Covered Charge in the Medical Benefits section of the certificate.

Charges that are due to complications of a non-covered service.

Charges that are incurred before the Covered Person's Effective Date or after the termination date of the coverage.

Free form text area

Therefore under the terms and conditions of the Certificate, coverage is not provided.

```
Active Content Group (Coverage Exclusions from Certificate)
    <Insert object here>
    Private Active Content(Behavioral Health or Substance Abuse)
    Private Active Content(Complications of a Pre-Existing Condition)
    Private Active Content(Non-covered Charges)
    Private Active Content(Complications for a non-covered service)
    Private Active Content(Prior to Effective Date or After Term Date)
        <Insert object here>
        Editable Container ()
            <Insert object here>
    Private Active Content(Free Form Text)
        <Insert object here>
        Editable Container ()
            <Insert object here>
            Paragraph (Free form text area...)
                <Insert object here>
            Paragraph (Charges that are incurred ...)
                <Insert object here>
```

4. Check that the interaction in your template design performs as required by using the P*review for EngageOne* option. See “Previewing an EngageOne template” on page 216 for more details.

▶ **To unmark text as editable within an editable container:** select the highlighted text you want to revert back to protected text. From the *Format* menu uncheck the *Mark as Editable* option,
– or –
select the *Mark as Editable* button on the *Standard* toolbar.
– or –
use CTRL+E.

Note that it is possible for an editable container to contain only non-editable text. If you delete an editable container, be aware that all content will be deleted within it, including protected, non editable text. If you want to delete the object in the logic map then you must move the content you still require outside the editable container and then delete the object.

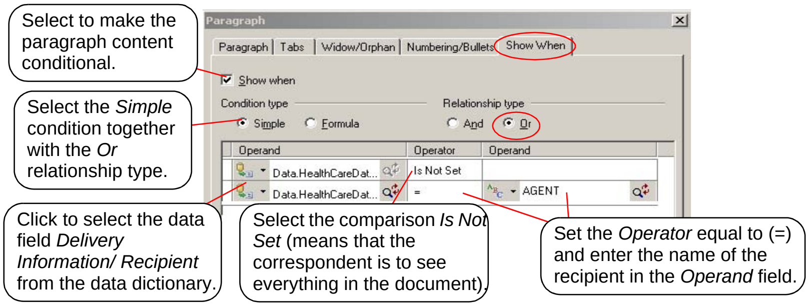
Recipient processing

When a template is submitted for delivery by the correspondent the delivery process is defined by the EngageOne Administrator, which involves defining and mapping devices and recipients to channels and grouping channels to form delivery options. This provides automatic carbon copy processing using recipients, meaning that the correspondent need only select one delivery choice and carbon copies will automatically be created and delivered.

Recipient conditions in the template are used to produce non-unique content. For example, if you do not want the user AGENT to receive the content designated to INSURED then you must check a condition against the recipient field for the content in your template design: if the recipient field *Is Not Set* (correspondent needs to see everything), *Or*, is equal to AGENT.

You must then use EngageOne Administration to set the *Inclusion Condition* in the delivery channel to match. See the Delivery Channels chapter in the EngageOne Administration Guide.

▶ **To set a recipient condition:** on the *Format* menu option click *Paragraph*. On the *Show When* tab of the *Paragraph* dialog box check the *Show When* option and enter the condition as required.



☑ Show when

Condition type: ● Simple ○ Formula

Relationship type: ○ And ● Or

Operand	Operator	Operand
Data.HealthCareDat...	Is Not Set	
Data.HealthCareDat...	=	AGENT

Select to make the paragraph content conditional.

Select the *Simple* condition together with the *Or* relationship type.

Click to select the data field *Delivery Information/ Recipient* from the data dictionary.

Select the comparison *Is Not Set* (means that the correspondent is to see everything in the document).

Set the *Operator* equal to (=) and enter the name of the recipient in the *Operand* field.

For more details about conditional paragraphs and creating conditional expressions see “Conditional paragraphs” on page 44.

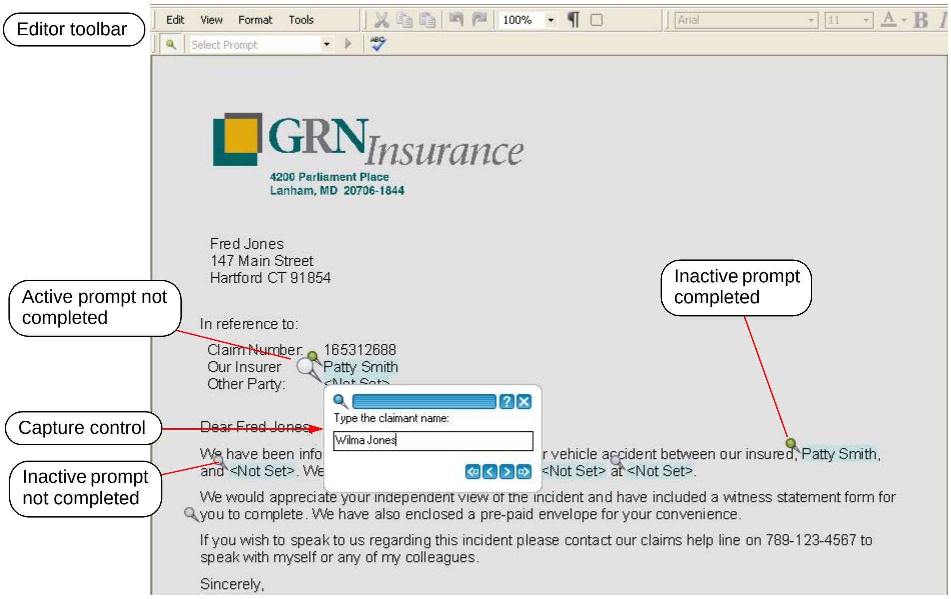
Previewing an EngageOne template

The *Preview for EngageOne* feature allows you to view a template as it would be seen by a correspondent in a front-office EngageOne environment. The most significant benefit of the preview feature is the opportunity to see how the user will interact with prompted input fields.

You must ensure that all data fields are mapped to your template before you can run a preview for EngageOne. This will be the interactive data definition (data map) produced from defining fields in the interactive data editor and generating the data dictionary.

▶ To run preview for EngageOne:

1. Select the *Design* tab and then the *Publications* bar.
2. Highlight the publication you want to preview.
3. From the *Tasks* menu select *Preview* for EngageOne.
4. A preview file is generated and the path is presented in the *Documents* window. Click on this path to review the output.



4200 Parliament Place
Lanham, MD 20706-1844

Fred Jones
147 Main Street
Hartford CT 91854

In reference to:

Claim Number: 165312688
Our Insurer Patty Smith
Other Party: <Not Set>

Type the claimant name:
Wilma Jones

Dear Fred Jones,

We have been info r vehicle accident between our insured, Patty Smith, and <Not Set>. We <Not Set> at <Not Set>.

We would appreciate your independent view of the incident and have included a witness statement form for you to complete. We have also enclosed a pre-paid envelope for your convenience.

If you wish to speak to us regarding this incident please contact our claims help line on 789-123-4567 to speak with myself or any of my colleagues.

Sincerely,

In interactive mode the document background is shaded and you can use the capture control to move through both the data prompts that have been introduced through the IDE and lists for selecting optional data defined using Active Content groups. See previewing IDE definitions on page 207 for details about using the capture control.

You can also tab through the prompts. When you reach the end of all data prompts included in the template the document background turns to white and turns back to grey again if you reach the first prompt in the document. The order in which you see the prompts is driven by the logic so if you want to change the sequence then you must change the positioning in the logic map.

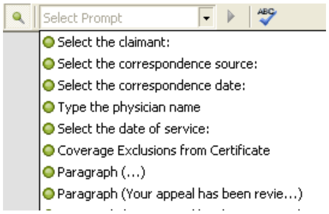
Any interactive data is identified on the document using pushpins. Green pushpins indicate data that has already been entered. White pushpins show positions where data input is still required. A larger version of pushpin indicates the current active prompt. This can be green or white depending on whether data has already been entered or not.

Editor toolbar

The **editor toolbar** provides standard text editing and formatting features for editable text and additional EngageOne related features. For details on how to use the editor toolbar features see “Editing features” on page 28 and “Creating paragraphs” on page 33. Pushpins can be toggled on and off using the *Pushpin* 🔍 button (only when the prompt box is closed).

> ⓘ Note that in a *Preview for EngageOne:*
> – You can only *Find and Replace* text in an editable text section. Also the *What to Search* options are not applicable.
> – Spell settings are not applicable when using the interactive editor. For more information about spell check management see the EngageOne Administration Guide.

You can select an interactive location from the drop-down list to display it directly for editing. Or you can click on any pushpin to display the related capture control/ editable text. The *Show Prompt* ▶ button will take you to the last interactive location selected.



Select Prompt

- Select the claimant:
- Select the correspondence source:
- Select the correspondence date:
- Type the physician name
- Select the date of service:
- Coverage Exclusions from Certificate
- Paragraph (...)
- Paragraph (Your appeal has been revie...)

If any font is used that is not installed on the correspondent’s machine, a warning message is issued when loading the template. The correspondent can suppress this message by selecting the *Do not show this again* check box and the nearest font substitute will be used. The correspondent can reactivate suppressed messages by selecting *Tools/Enable suppressed messages* from the editor toolbar.

Modifying sample data

If your sample data does not allow you to preview extensively enough, for example, if you want to include some sample data for repeating groups (previewing tables) then you can extract the sample data and modify it as follows:

▶ To modify sample data in order to preview repeating group entries:

1. In the Work Center Manager select the *Data* tab and then click the *Sample Data* bar.
2. Highlight the file you want to work with and from the *Tasks* menu select *Extract Sample Data*.
3. Modify the repeating data in the XML file using a text editor such as Notepad.

4. In *Publications* view right-click on *Publication* in the logic map and select *Properties* from the short cut menu.
5. Click the *Sample Data* browse button.
6. Click *My Computer* and select the file you modified.

Publishing for EngageOne

Publishing for EngageOne Interactive differs from regular DOC1 publishing in that a template package is created and saved to a location of your choice. This package comprises an EngageOne template manifest that describes all the other files that make up the template. For details of how to publish an EngageOne Interactive template see “Publishing and deployment” on page 329. EngageOne Administration handles the import of these template packages into the EngageOne interactive environment where they are made available to the correspondent via a Web client.

Additionally, Active Content can be published as a group or individually. These files are also imported into the EngageOne environment. For more detailed information about this process and configuring the EngageOne production environment see the EngageOne Administration Guide. Take special care when publishing Active Content on its own – see “Using Active Content to update publications in production” on page 233.

In some cases you might need to make a wider selection of fonts available to the correspondent within editable sections of text, other than those contained within the template design. This is handled in the Publishing Wizard, where, when publishing for *Interactive* you can select additional fonts for the correspondent to use in the *Additional Resources* page. Note that the fonts specified will not be available in preview mode and you must still use the *Publish Resources for EngageOne* option as usual.

All data fields that have been used in the template must be linked with actual data elements. With EngageOne these references are resolved by referencing the data map produced from the interactive data file. See “Generating a dictionary and data map for IDE” on page 210

Publishing resources for EngageOne

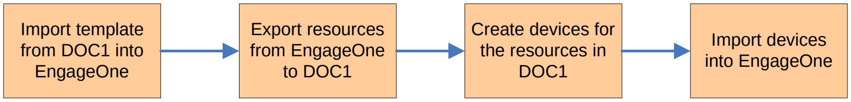
You must publish resources independently for EngageOne templates as they are not included in the EngageOne publishing process as with regular DOC1 publications.

ⓘ NOTE THAT WHEN YOU *PUBLISH RESOURCES FOR ENGAGEONE* THIS INCLUDES PUBLISHABLE ACTIVE CONTENT AS WELL AS FONT AND IMAGE RESOURCES.

When the completed template is imported into EngageOne, resources for that template are gathered by querying all the templates for the resources in the list. Using the *Delivery Management/Resources/Export Resources* option in EngageOne Administration you can export template resources to a local file in XML format, which is passed manually to DOC1 Designer.

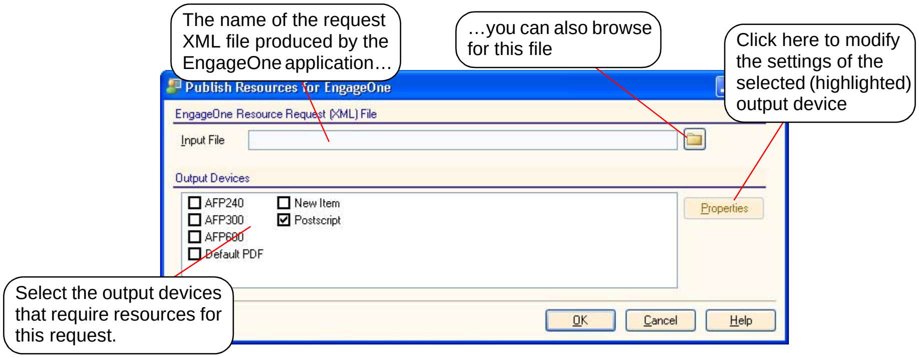
The *Publish EngageOne Resources* option in DOC1 Designer allows the input file to be browsed and read and then prompts for the target output device definitions to publish to. A single zip file is created that contains all the resources for the selected devices. This zip file is then imported into EngageOne Administration to create or update all devices installed on the server.

The basic process flow is illustrated in the following diagram.


Import template from DOC1 into EngageOne
Export resources from EngageOne to DOC1
Create devices for the resources in DOC1
Import devices into EngageOne

⯈ To publish resources for EngageOne:

1. In the Work Center Manager window select the *Design* tab and then the *Publications* bar.
2. Highlight the template to be published.
3. From the *Tools* menu select *Publish Resources for EngageOne.* This displays the dialog for selecting the output devices required.



The name of the request XML file produced by the EngageOne application…

…you can also browse for this file

Click here to modify the settings of the selected (highlighted) output device

Select the output devices that require resources for this request.

Publish Resources for EngageOne

EngageOne Resource Request (XML) File

Input File

Output Devices

□ AFP240
□ AFP300
□ AFP600
□ Default PDF
□ New Item
☑ Postscript

4. Click *OK*. Using the Explorer dialog provided, name and save the file to the location specified by your EngageOne Administrator. A file package is created ready for importing into the EngageOne Server. For more information about delivery management see the EngageOne Administration Guide.

Controlling the output environment

This section discusses features that allow you to issue specific commands to the intended output device within the datastream created by DOC1 Generate.

These options are only used where dynamic changes to the output environment are required. General settings for the intended printer/browser are included in an output device object that is specified as part of the *Publish* task.

Spot color printers

Full color printers are able to mix toners to allow a full range of colors to be printed. However some printers only support the use of black plus one or more other single colors from fixed toner supplies. This is known as spot or highlight color.

When selecting colors in DOC1 for use with spot color printers you will need to be aware how your printer interprets color commands that are passed to it within an output datastream. If the printer supports only a single spot color this will typically be used whenever a color other than black is referenced within the datastream. Printers that support multiple spot colors will typically attempt to map colors within the datastream to the best match in their toner palette.

Within the DOC1 Designer the use of spot color commands in the output datastream is specified by using a **highlight color** group in application logic. All presentation objects within such a group (other than charts and images) will be assigned the defined color. Existing color specifications in these objects will be overridden.

Objects not within a highlight color group will use their regular color settings but will be assumed to be black in a spot color environment.

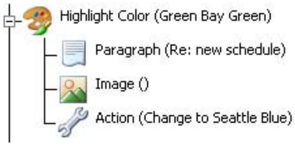
The initial color to be used is specified as part of the highlight color object. If your intended printer supports multiple spot colors you can use an Action to change the active color. The selected color stays in force for objects within highlight color groups until a further change is requested.

If you intend to publish to an AFPDS printer that supports highlight color then you will need to specify the Highlight Color Space (HCS) options. Refer to the IBM MO:DCA specification for full details of the settings.

▶ To create a highlight color group:

1. Mark the insertion point in the logic map.

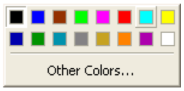
   Bear in mind that highlight color is a group object that contains the presentation objects to be colored.



Highlight Color (Green Bay Green)
Paragraph (Re: new schedule)
Image ()
Action (Change to Seattle Blue)

2. From the menu select *Insert/Tool Objects/Highlight Color*.
3. In the *Highlight Color* dialog box select the spot color to be applied to the grouped objects.

   A 16 color palette is used to represent the typical options of a spot color environment. For other requirements use the *Other Colors* option.

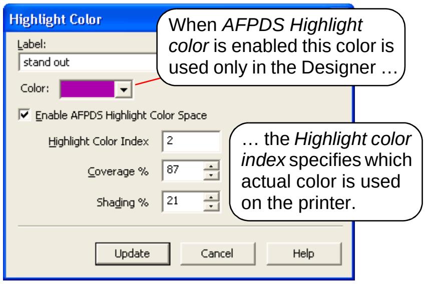


4. Enable AFPDS Highlight color printing if required.

   Highlight color index is
   – either an index to a Color Mapping Table (CMT) when *coverage* and *shading* are both set to zero
   – or is a number associated with the printer’s highlight color.

   Coverage defines the amount of coverage or density of ink that is used to render the output. The default value is 100%.

   Shading defines the mixing of the default color (usually black) and the highlight color to render a mix of the two. The default value is 0%.



Highlight Color

Label: stand out

Color:

☑ Enable AFPDS Highlight Color Space

Highlight Color Index: 2

Coverage %: 87

Shading %: 21

When *AFPDS Highlight color* is enabled this color is used only in the Designer …

… the *Highlight color index* specifies which actual color is used on the printer.

5. Add objects to the group.

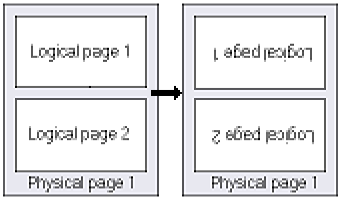
   Create logic map objects in the normal way. If the objects to be colored already exist you can drag them into the group.

▶ To change the active spot color: create an action of type *Change Highlight Color* and specify the new color in the *Action* dialog box. This action is only available within a highlight color group.

Controlling printer features

Some printers use multiple paper trays to allow dynamic selection of the paper type to be used or multiple output bins to collate the pages generated into a custom sequence.

Additionally, some printers also allow printing on both sides of the physical sheets of paper. This is known as *duplex* printing. When it is decided to print single-sided only, as *simplex*.



Some printers also provide *tumble* printing – i.e. turning the page around top to bottom so that it is printed the other way up.

Where these features are supported, DOC1 includes instructions for controlling the printer into the output datastream being generated.

The initial settings for these features are specified as part of the output device options that are used when publishing.

SEE T"OUTPUT, MEDIA AND RESOURCES" ON PAGE 274 FOR DETAILS.

To switch the settings during the processing of a job you need to include Action objects in the appropriate part of your document logic.

When creating an action to switch trays or bins you need to be aware how they are referenced on your target printer. DOC1 provides a set list of trays and bins that you can map using the appropriate format for the target printer. The mappings for the trays/bins are set in the Output Devices bar in the Environment tab. You need to be aware of the mappings to select the correct tray/bin when you insert an Action object. While you can use an action to override which tray/bin is used when the document is published, you cannot override the tray/bin mappings created for the output device.

▶ To create a printer control action:

1. Mark the insertion point in the logic map.

   The change is made in relation to the page being generated when the action is called.

2. From the *Insert* menu select *Control Objects/Action*.

3. In the *Action* dialog box choose a type of
   Select input tray,
   Select output bin or
   Simplex/Duplex.

4. Specify when the change is actioned.

   By default, the change will take place immediately – i.e. the instruction to change the tray or bin will be included as part of the page currently being generated by the job. If you want to delay the change until the next new page select the *Active on next page* option.

PostScript DSC

DSC comments are typically used to communicate with document manager systems in a PostScript environment.

PostScript output has specific sections that provide controls related to pages, documents or the entire job. You will need to indicate what section your DSC comment is to appear in.

The comment generated is not validated by DOC1 and the user must ensure that it conforms with the DSC guidelines published with your document manager. Invalid comments may result in unprintable PostScript files.

▶ To program a DSC comment:

1. Mark the insertion point as required in the logic map and from the menu select *Insert/Control Objects/Action*.
2. In the *Action* dialog box choose a type of *PostScript DSC*.
3. Use the *DSC comment positioning* options to indicate the PostScript section to which the comment will be applied and use *Value* to specify the comment.

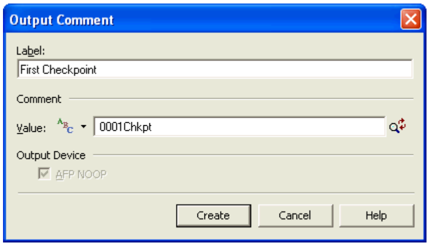
Adding output comments

Output comments are typically used as markers to control any post processing you may want to apply to your output datastream. When working with AFP output, these comments are inserted as No Operation (NOOP) instructions in the generated output. Comments can be applied at both publication and document levels.

ⓘ NOTE THAT OUTPUT COMMENT ENTRIES ARE IGNORED WHEN GENERATING OUTPUT OTHER THAN AFP.

▶ To include comments in the generated output datastream:

1. In the logic map mark an insertion point where you want the output comment to appear.
2. Click the toolbar button
   – or –
   from the *Insert* menu choose *Tool Objects and click on the Output Comment option*.
3. Enter details about the comment in the *Output Comments* dialog box as shown.



Output Comment

Label: First Checkpoint

Comment

Value: 0001Chkpt

Output Device

☑ AFP NOOP

Using Active Content

Active Content objects contain segments of document content and logic. They provide a highly flexible way to share and reuse such segments across multiple publication designs.

- In its simplest form Active Content provides a means to make the same segment of document design available to many different publications.
- By using parameters and return values Active Content can be used to create reusable functions.
- When used as part of an Active Content group these files can be conditionally linked via property settings to provide a mechanism for dynamically building frequently reused sequences of content.
- If you are designing document templates for use with the EngageOne environment you can present a list of optional content for end user selection in the EngageOne interactive editor using Active Content groups.
- Active Content can be published independently and then used to dynamically update one or more publications that are already in production.
- Active Content can also be called as a keyed object. This allows specific Active Content objects to be called depending on the value of a key usually set by input data. See “Keyed objects” on page 158 for details.

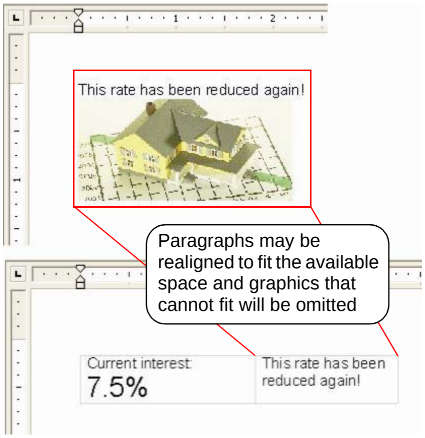
Creating a basic Active Content object

By creating an Active Content object you are making document content and logic that can be inserted wherever it is needed in other design elements. This is sometimes called public Active Content. You could create an Active Content object that organizes customer details and presents them in a consistent fashion. You could reference that piece of Active Content wherever you wanted customer details to appear in a document layout.

Within the Active Content editor, paragraphs are created and manipulated exactly as you would within the main document editor and you can make use of all text related features. You can also include graphics (shapes or images) but note that these must be *in-line* or *anchored* to a paragraph; you cannot used a *fixed* anchor type.

Active Content can be referenced anywhere in other design elements but the content they provide may be automatically adjusted to fit the available space. To help with this you may want to use the *Page Setup* options in the Active Content object to adjust the available workspace to emulate the dimensions of a particular area such as a text box or table cell.

IMPORTANT: if you intend to make Active Content independently publishable, you cannot include data fields directly. For more information see “Using Active Content to update publications in production” on page 233.


This rate has been reduced again!
Current interest
7.5%
This rate has been reduced again!

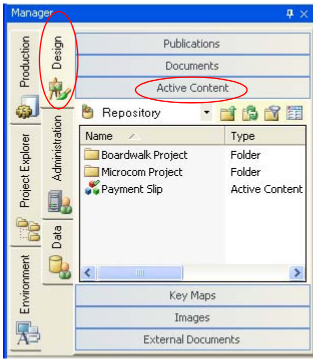
Paragraphs may be realigned to fit the available space and graphics that cannot fit will be omitted

Otherwise, when working with Active Content you are able to select fields from any data dictionary known to the repository but if you intend to share Active Content across multiple publications you should consider carefully how the fields you use will be mapped to data elements in the actual DOC1 input streams if these do not always share the same format. Note that where a data map and sample data, (or interactive data model in the case of EngageOne templates) have already been assigned to a publication these will also be made available to an Active Content object when it is launched from within a publication. See “Sharing and synchronization” on page 101 for more information.

Existing Active Content objects can be edited either independently using the Active Content editor in the Work Center Manager or, within the context of a parent document.

▶ To create an Active Content object:

1. In the Work Center Manager select the *Design* tab and then click the *Active Content* bar. From the *File* menu select *New Active Content*.
2. A new Active Content object is displayed. Enter a name and press *Return*.
3. The *Active Content* dialog box is displayed.
   For basic requirements simply leave the *Label* name in the *Properties* tab, or overtype with a new one and click *OK*.
4. The Active Content editor is launched.
   Add logic and content in the normal way.
5. Save and issue the new file.
   Note that Active Content objects must be issued before they can be used within document designs.



Name	Type
Boardwalk Project	Folder
Microcom Project	Folder
Payment Slip	Active Content

What the Active Content dialog tabs are used for:

Properties Properties are used in conjunction with Active Content groups to conditionally select the appropriate Active Content objects according to the values specified. For more information see “Active Content groups” on page 230. You can also enable Active Content to be published independently – see “Using Active Content to update publications in production” on page 233.

Parameters This tab allows you to specify the expected parameters when using Active Content as a function. See the following section “Using Active Content as a function” for more details.

Languages If you intend to use localized paragraphs within the file use this tab to set the supported languages. See “Localization” on page 183 for details.

Page Setup Use this tab to specify the size of the Active Content layout space when you want to emulate a specific area in which the content will need to fit.

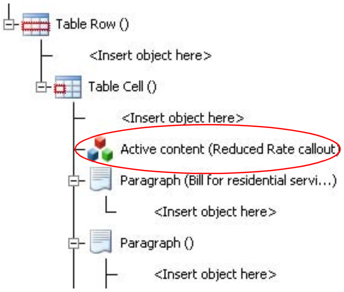
Sample data As with a regular publication design you may want to use sample data to see how it will affect the Active Content logic and layout. When editing Active Content from within a parent document such data assignments for the publication are automatically passed down to file being loaded. At other times you can use this tab to assign the required sample data and data map. See “Sample data and data mapping” on page 177 for more information about these resources.

For details about the *System Properties* pane on the *Active Content* dialog box see “Using Active Content to update publications in production” on page 233.

▶ **To update the attributes of an Active Content object:** open the file from the Work Center and then double-click the main Active Content object.

▶ **To reference Active Content in a design element:**

1. Ensure that the required Active Content object has been issued.
2. Open the publication/document if it is not already loaded.
3. Position the cursor at the paragraph any new content is to follow or simply mark the appropriate insertion point in the logic map.
4. From the *Insert* menu click *Control Objects/Active Content*.



5. In the *Active Content* dialog box give the reference a *Label* and use the folder button 📁 to search for the required object.
6. Click *Create* to add the reference and incorporate the logic and content into the active design.

Using Active Content as a function

PARAMETERS AND RETURN VALUES ARE NOT SUPPORTED WHERE ACTIVE CONTENT IS CALLED AS A KEYED OBJECT. SEE "KEYED OBJECTS" ON PAGE 158 FOR DETAILS OF THIS FEATURE.

By making use of input parameters and the return value you can use Active Content objects to create repeatable logic functions that can be called as required by other design elements. For example, you may have an Active Content function that calculates the type of tariff applicable to various transaction amounts. The function would receive the amount as a parameter, evaluate it using a case object and return a string that indicates the type of the tariff applied. You could call the Active Content function wherever you needed to perform this evaluation. Note that Active Content may often be used as both a function and to provide content to the design element that references it.

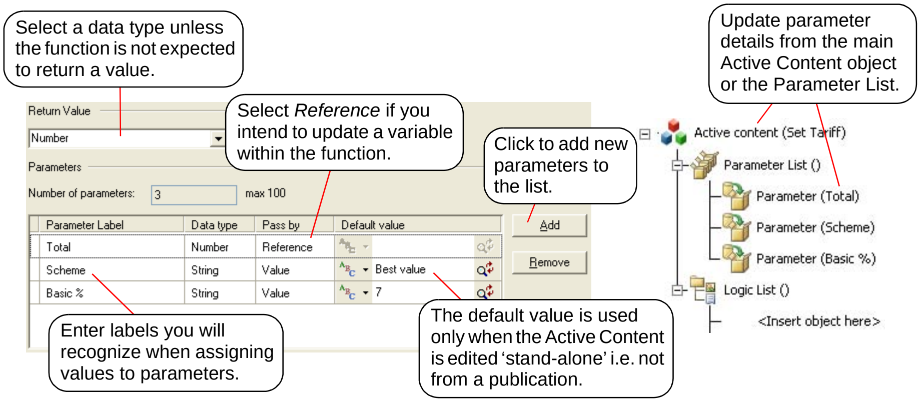
Any value type can be passed as a parameter to Active Content and these can be used as you would use other values in both logic and layout. Parameters can also include references to other Active Content objects effectively allowing you to nest function calls if required.

Where variables or other Active Content functions are used to provide parameters you can also dynamically update their values if the *Pass by Reference* option is set.

▶ To specify a return value and parameters for an Active Content object:

1. Double-click on the main Active Content object in the logic map to open the *Active Content* dialog box.

2. Use the *Parameters* tab to specify the type of return value and each of the parameters that are expected.



Return Value: Number

Parameters

Number of parameters: 3 max 100

Parameter Label	Data type	Pass by	Default value
Total	Number	Reference	
Scheme	String	Value	Best value
Basic %	String	Value	7

Select a data type unless the function is not expected to return a value.

Select *Reference* if you intend to update a variable within the function.

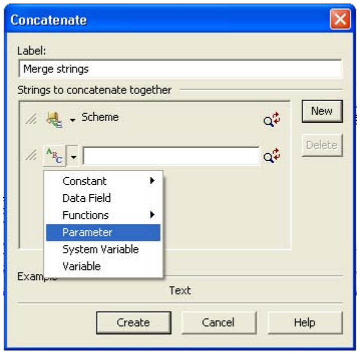
Click to add new parameters to the list.

Update parameter details from the main Active Content object or the Parameter List.

Enter labels you will recognize when assigning values to parameters.

The default value is used only when the Active Content is edited 'stand-alone' i.e. not from a publication.

▶ To pass parameters to an Active Content function: double-click the Active Content object within a design element. The *Active Content* dialog box will allow you to assign parameter values.



Concatenate

Label: Merge strings

Strings to concatenate together

Scheme

Example

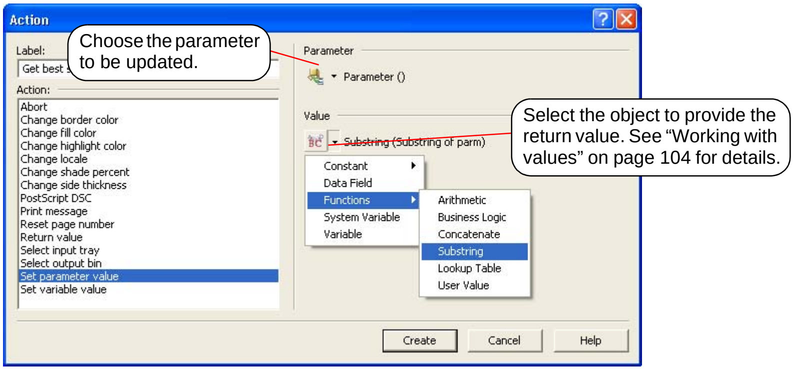
Text

▶ To use parameters within Active Content logic: select *Parameter* from the value types when specifying input to any object such as a variable or standard function.

▶ To use parameters directly within a paragraph: select the appropriate insertion point within a paragraph and from the *Insert* menu click *Field/Parameter.*

▶ To update a Pass By Reference parameter:

1. Within the Active Content object, mark the insertion point in the logic map where the value required to update the parameter is available.
2. From the *Insert* menu select *Control Objects/Action.*
3. In the *Action* dialog box select an action type of *Set parameter value.*



Action

Label: Get best s

Action:

Abort
Change border color
Change fill color
Change highlight color
Change locale
Change shade percent
Change side thickness
PostScript DSC
Print message
Reset page number
Return value
Select input tray
Select output bin
Set parameter value
Set variable value

Parameter: Parameter ()

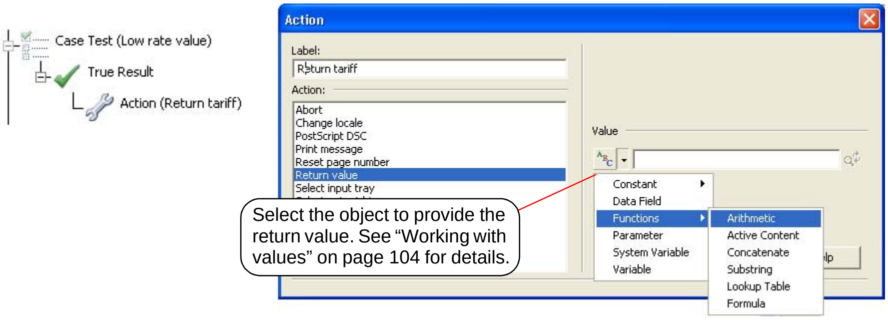
Value: Substring (Substring of parm)

Choose the parameter to be updated.

Select the object to provide the return value. See “Working with values” on page 104 for details.

▶ To return a value to the calling object:

1. Within the Active Content object, mark the insertion point in the logic map where the return value will be available.
2. From the *Insert* menu select *Control Objects/Action*.
3. In the *Action* dialog box select an action type of *Return value*.



Case Test (Low rate value)
True Result
Action (Return tariff)

Action

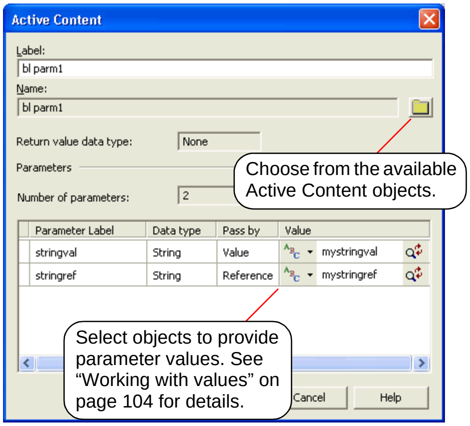
Label: Return tariff

Action: Return value

Select the object to provide the return value. See “Working with values” on page 104 for details.

▶ To call an Active Content function that returns a value:

1. Create the object that will receive the value (*Variable*, *Action*, *Condition*, etc.).
2. In the appropriate object dialog box select a value type of *Function/Active Content*.
3. In the *Active Content* dialog box identify the required function and specify any parameters.



Active Content

Label: bl parm1

Name: bl parm1

Return value data type: None

Parameters

Number of parameters: 2

Parameter Label	Data type	Pass by	Value
stringval	String	Value	mystringval
stringref	String	Reference	mystringref

Choose from the available Active Content objects.

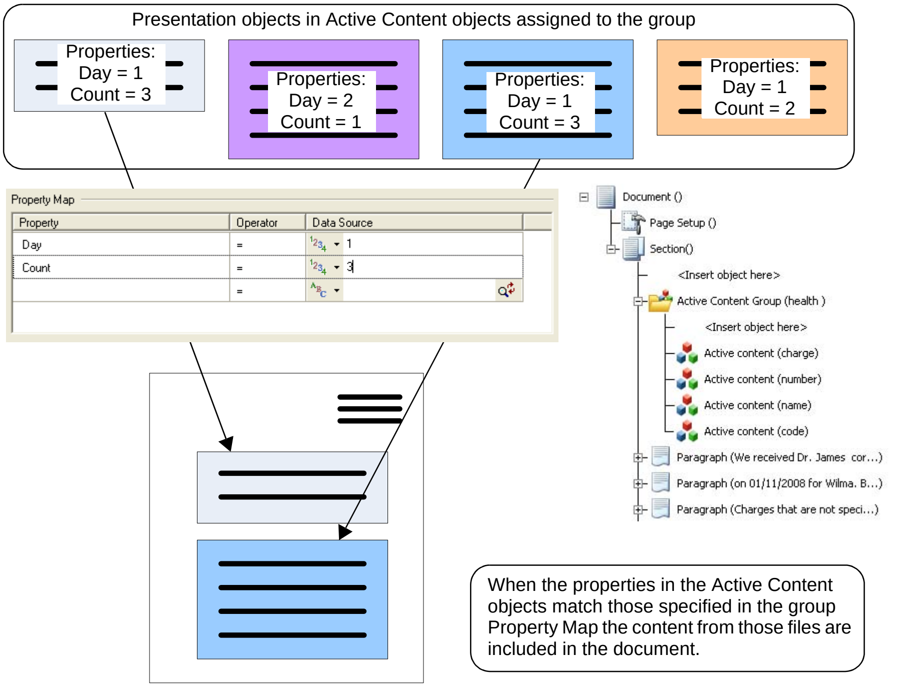
Select objects to provide parameter values. See “Working with values” on page 104 for details.

Active Content groups

Active Content groups allow you to build linked but variable groups of pre-defined content. For example, you may want to use an Active Content group to build the body of a customized letter from a range of standard paragraphs or you might want to create a variable ‘terms and conditions’ section within a document design.

When working with Active Content groups you need to ensure that each segment of content (usually one or more paragraphs) are included in separate Active Content objects. These files are then referenced in the Active Group in the required sequence. Variability is provided in relation to the **properties** specified for each Active Content object: only those files that match the group’s property map will be selected. If required you can make an entire Active Content group conditional.

WHEN DESIGNING FOR ENGAGEONE APPLICATIONS YOU CAN ALSO SPECIFY PROPERTIES AS INTERACTIVE PROMPTS. SEE “USING ACTIVE CONTENT WITH ENGAGEONE” ON PAGE 234 FOR DETAILS.


Presentation objects in Active Content objects assigned to the group
Properties: Day = 1 Count = 3
Properties: Day = 2 Count = 1
Properties: Day = 1 Count = 3
Properties: Day = 1 Count = 2

Property Map

Property	Operator	Data Source
Day	=	1
Count	=	3
	=	

Document ()
Page Setup ()
Section()
<Insert object here>
Active Content Group (health)
<Insert object here>
Active content (charge)
Active content (number)
Active content (name)
Active content (code)
Paragraph (We received Dr. James cor...)
Paragraph (on 01/11/2008 for Wilma. B...)
Paragraph (Charges that are not speci...)

When the properties in the Active Content objects match those specified in the group Property Map the content from those files are included in the document.

Unlike other uses of Active Content when working with groups you may want to create Active Content objects exclusively for use by a single document design. These are known as **private Active Content**.

▶ To create an Active Content group:

1. Create or update the public Active Content objects that will make up the group.
   Add suitable properties to each. If you are using custom properties ensure that they share the same name and have values assigned.

2. Create the Active Content group.
   In the design element that will include the group, mark the insertion point as required and from the *Insert* menu select *Control Objects/Active Content Group*
   – or –
   click the button in the toolbar.

▶ To add public Active Content object references to the group:

1. Position the cursor at the insertion point under the appropriate Active Content group.

2. From the *Insert* menu click *Control Objects/Active Content/Active Content*
   – or –
   click the button in the toolbar.

3. In the *Active Content* dialog box give the reference a *Label* and use the folder button to search for the required object.

▶ To add private Active Content to the group:

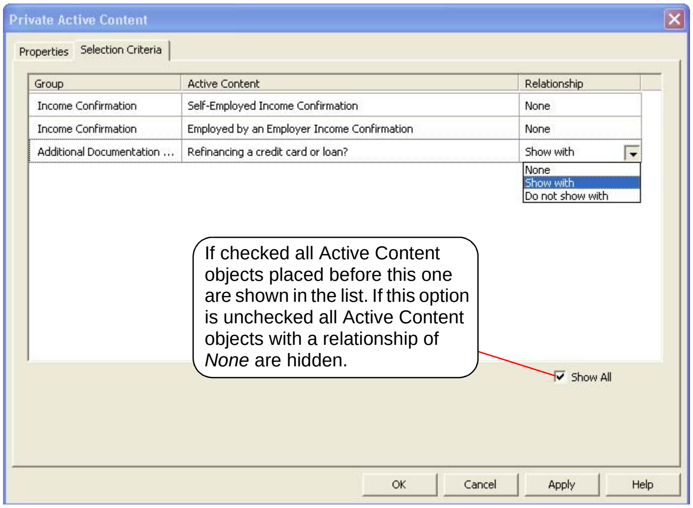
1. Position the cursor at the insertion point under the appropriate Active Content group.

2. From the *Insert* menu click *Control Objects/Active Content/Private Active Content*
   – or –
   click the button in the toolbar.

3. The *Private Active Content* dialog is displayed.

4. If you want to define properties that should be matched, use the *Add* button from the *Properties* tab to define custom properties for private Active Content. If the properties do not match the content will not be included in the document and the object will be grayed out in the logic map.

5. The *Selection Criteria* tab shows all the Active Content groups and group entries that precede this active group entry. For each entry in the list one of three relationships can be established, which determines inclusion:
   None – the default value and there is no relationship
   Show with – the Active Content is only placed in the document if the properties match AND the selected Active Content has been shown.
   Do not show with – the Active Content is only placed in the document if the properties match AND the selected Active Content has NOT been shown.



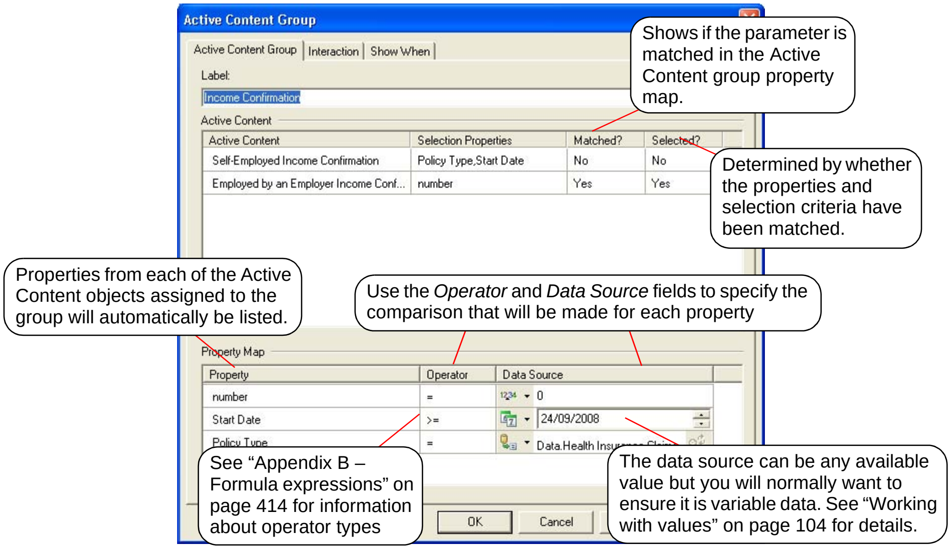
Group	Active Content	Relationship
Income Confirmation	Self-Employed Income Confirmation	None
Income Confirmation	Employed by an Employer Income Confirmation	None
Additional Documentation ...	Refinancing a credit card or loan?	Show with

☑ Show All

If checked all Active Content objects placed before this one are shown in the list. If this option is unchecked all Active Content objects with a relationship of *None* are hidden.

The Active Content Group dialog shows the Active Content entries within the group, the match status and whether the entry has been selected for inclusion. The associated properties are taken over from the Active Content group entry but can be changed in the Property Map for the group.

▶ **To map properties associated with the Active Content to a data source:** double-click on the required Active Content group object in the logic map. The Active Content Group dialog is displayed.



Label: Income Confirmation

Active Content	Selection Properties	Matched?	Selected?
Self-Employed Income Confirmation	Policy Type,Start Date	No	No
Employed by an Employer Income Conf...	number	Yes	Yes

Property	Operator	Data Source
number	=	0
Start Date	>=	24/09/2008
Policy Type	=	Data.Health Insur

Shows if the parameter is matched in the Active Content group property map.

Determined by whether the properties and selection criteria have been matched.

Properties from each of the Active Content objects assigned to the group will automatically be listed.

Use the *Operator* and *Data Source* fields to specify the comparison that will be made for each property

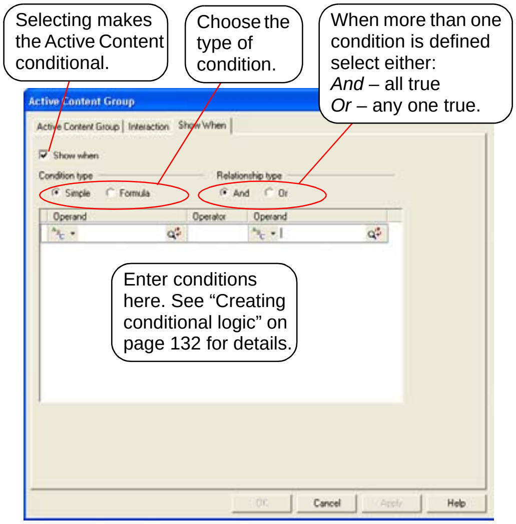
See “Appendix B – Formula expressions” on page 414 for information about operator types

The data source can be any available value but you will normally want to ensure it is variable data. See “Working with values” on page 104 for details.

Active Content groups can be made conditional so that they appear when certain criteria are met, for instance, you may want to show a group only if a customer has a certain type of insurance. You will typically want to use values from data fields and variables that have been made available to the template as part of the condition.

Note that if you make an Active Content group conditional it will not be shown in the editor if the specified condition evaluates to *false*. To change the settings you will need to use the logic map to work with the relevant Active Content object. See “Using the logic map” on page 92 for more information.

▶ **To make the entire Active Content group conditional:** double-click the Active Content group object to display the *Active Content Group* dialog. On the *Show When* tab of the *Active Content Group* dialog box check the *Show When* option and enter the condition as required.



Selecting makes the Active Content conditional.

Choose the type of condition.

When more than one condition is defined select either:
And – all true
Or – any one true.

Enter conditions here. See “Creating conditional logic” on page 132 for details.

Active Content Group

☑ Show when

Condition type: ● Simple ○ Formula

Relationship type: ● And ○ Or

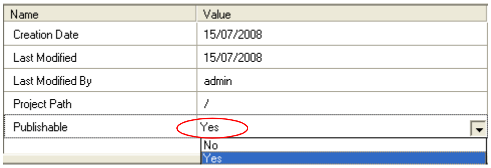
Using Active Content to update publications in production

Active Content can also be published independently. This allows you to update publications dynamically that are already in production. You can enable and disable this feature for each Active Content object.

Use this option with caution as the impact on production jobs may be significant. For more information about publishing Active Content see page 329

IMPORTANT: data fields cannot be used directly in Active Content that is published independently. You can however pass them as parameters to the Active Content – see “Using Active Content as a function” on page 227.

▶ To enable independent publishing of Active Content:



Name	Value
Creation Date	15/07/2008
Last Modified	15/07/2008
Last Modified By	admin
Project Path	/
Publishable	Yes

1. In the Work Center Manager select the *Design* tab and then click the *Active Content* bar.
2. Open the Active Content properties.

   From the logic map right-click the Active Content object and select *Properties*.
3. In *System Properties*, click on the *Publishable* value, set it to *Yes* and click OK. Note that you cannot set this publishable property to *Yes* if data associated with it is already being used.
4. Save and issue the Active Content.

Publishable Active Content will be included when you publish resources for EngageOne. For more information see “Publishing resources for EngageOne” on page 218.

Note that the Active Content HIP file is not created when you publish the publication. You must publish these separately – see “Running a publishing task” on page 336.

Using Active Content with EngageOne

Active Content objects play a special role in EngageOne applications. Typically an EngageOne document template will contain Active Content groups to present list(s) of optional content to the correspondent. Be aware to check the *Interactive* box on the *Interaction* tab of the Active Content group dialog – see “Interactive settings” on page 234 for more details. Depending on the choices made other content could be included later in the document or another list could be activated.

As an EngageOne solution commonly includes a large number of templates it will also often make use of independently publishable Active Content to avoid the need to republish all the templates when minor content changes are required.

For more detailed information about using interactive optional content using Active Content groups you should refer to the Active Content and EngageOne section in “Creating EngageOne templates” on page 201.

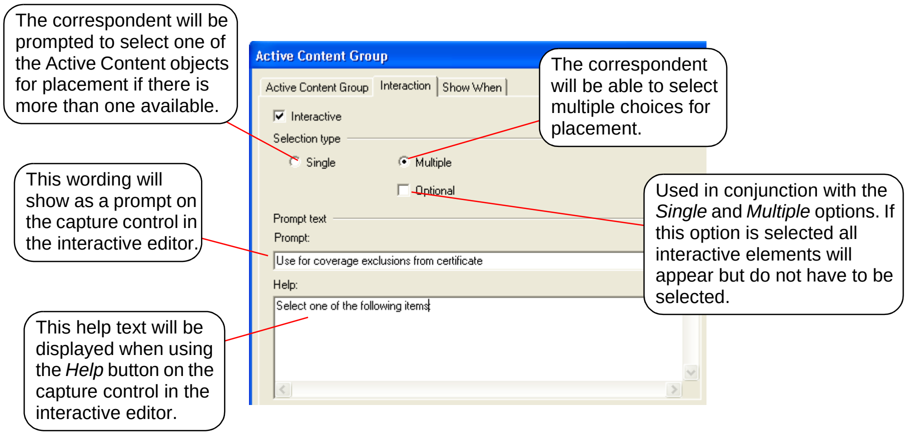
Interactive settings

The *Interaction* tab of the *Active Content Group* dialog is used to specify that the template is to be used in interactive mode in the EngageOne front-office environment and allows you to set particular interactive characteristics for display in the interactive editor.

▶ To make an Active Content group interactive:

1. Create the Active Content group as before – see page 231.

2. Click on the *Interaction* tab of the *Active Content Group* dialog.



☑ Interactive

Selection type

○ Single ● Multiple

□ Optional

Prompt text

Prompt: Use for coverage exclusions from certificate

Help: Select one of the following items

The correspondent will be prompted to select one of the Active Content objects for placement if there is more than one available.

The correspondent will be able to select multiple choices for placement.

This wording will show as a prompt on the capture control in the interactive editor.

Used in conjunction with the *Single* and *Multiple* options. If this option is selected all interactive elements will appear but do not have to be selected.

This help text will be displayed when using the *Help* button on the capture control in the interactive editor.

Identifying Input Data

About DOC1 input data

Input data is the variable information you use as part of your documents – typically an extract from your corporate databases. DOC1 supports input data in the following formats:

Keyed records – effectively a raw data extract where publication data is made up of one or more record types. Each record must be identifiable by the contents of a key field.

Delimited – a variation of keyed records where field locations are predefined using one or more consistent delimiter characters. This is sometimes known as a CSV (comma-separated values) file.

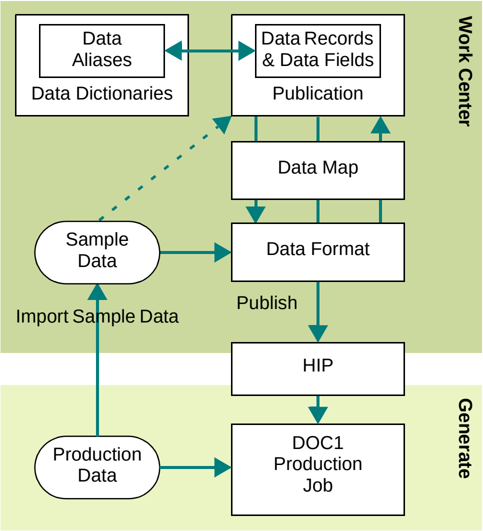
XML – any well formed XML file with an associated DTD or Schema.
Note that XML support is an optional extra and is only enabled to users who install both the DOC1 Work Center and Generate with the appropriate license keycodes.

The type and structure of the input data file expected by a particular DOC1 production job is described in a **data format** object. A data format is initially created by importing a sample of the expected input data which is then used as a template for identifying the expected records and fields. This is known as a **sample data** file.

Once you have imported your sample data into a new data format you can then use the Data Format Editor (DFE) to specify and corroborate the details of the data elements it contains. The type of work you need to do in the DFE will depend on the type of input data you are working with.

When you are designing documents you work with indirect references to fields and records. These references are stored as part of a **data dictionary** and you can use such dictionaries to represent the data available for a particular publication, document or any other logical grouping you want to establish.

Before being published all the data aliases used by a publication must be linked with actual elements defined in the data format. This is done by creating a a **data map** object. This can be done as part of the publication process or you can create one in advance.



Depending on your working practices you may want to create a data format first and then use it to export a data map and a data dictionary. This will allow document designers to work with a list of known data elements and avoids the need to manually link individual elements.

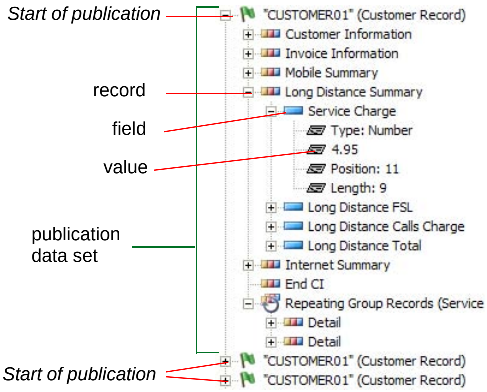
If you are working with EngageOne, document input files are handled differently. For more details see “EngageOne data” on page 203.

Once you have associated a data map with a publication you will be able to use the sample data to provide extended example data for data aliases used in document designs. You will also be able to switch between sample data files providing they conform to the structure known to the data format referenced by the data map. You may want to do this for instance, if you may have a range of files that contain a variety of data samples.

Data structure

In the production environment DOC1 Generate requires input data to be supplied in a single file for each job. You may need to merge multiple input sources using the PBBI Data Flow product suite or similar.

Regardless of the type of input data being used DOC1 sees it as a grid of data elements consisting of one or more **record** types which themselves consist of one or more **fields**. A field is the data element that provides an actual **value** to DOC1.



"CUSTOMER01" (Customer Record)
Customer Information
Invoice Information
Mobile Summary
Long Distance Summary
Service Charge
Type: Number
4.95
Position: 11
Length: 9
Long Distance FSL
Long Distance Calls Charge
Long Distance Total
Internet Summary
End CI
Repeating Group Records (Service
Detail
Detail
"CUSTOMER01" (Customer Record)
"CUSTOMER01" (Customer Record)

The data intended for an individual publication must be provided in one or more sequential records within the input data file. This is known as a **publication data set**. It follows that the number of publications produced by a job depends on the number of publication data sets in the input data file.

The start of a new publication data set within the input data is indicated by the presence of a specific data element known as a **start of publication** record. Such records tell DOC1 to expect a new publication data set and to reload the appropriate logic. At least one element within a data format must be assigned this flag.

A job can produce multiple types of publication. DOC1 Generate decides which design to apply to each publication depending on the *start of publication* record it encounters.

Each publication data set is discreet – i.e. you cannot directly share data between publications. However, document designs can include variables of job scope to pass values if required.

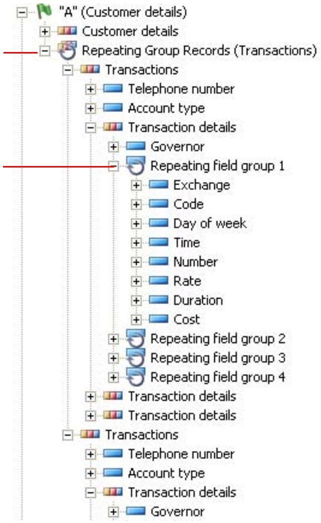
See “Variables” on page 110 for details of storing and passing values.

Repeating data

Repeating data is the general term used for data elements or groups of elements that appear iteratively within the input file. These typically contain transactional information such as the details of an itemized phone bill or a list of purchases on a credit card.

If you are working with XML input, any repeating data structures will be distilled automatically from the sample data provided.

For keyed or delimited data you will need to specifically identify repeating data structures. Both records and fields can repeat both individually or as sequential groups of repeating elements.



start of a group of 2 repeating records (transaction summary & transaction details)

start of a group of 8 repeating fields

- "A" (Customer details)
  - Customer details
  - Repeating Group Records (Transactions)
    - Transactions
      - Telephone number
      - Account type
      - Transaction details
        - Governor
        - Repeating field group 1
          - Exchange
          - Code
          - Day of week
          - Time
          - Number
          - Rate
          - Duration
          - Cost
        - Repeating field group 2
        - Repeating field group 3
        - Repeating field group 4
      - Transaction details
      - Transaction details
    - Transactions
      - Telephone number
      - Account type
      - Transaction details
        - Governor

See “Defining repeating data structures” on page 255 for more information.

Types of input data

Keyed records

This type of input is effectively raw data although it must have a recognizable record structure. Each record type must be uniquely identifiable by the contents of a **key field**.

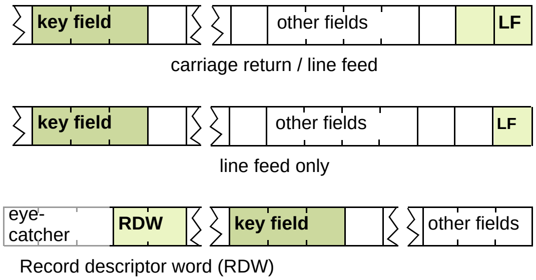
The key field can appear anywhere within a record but must be in the same position and have the same length in every record type. It typically contains a string that acts as a code for the relevant record type.

Various methods are supported for delineating the records within the input data file:

Any part of the remaining data within individual records can be defined as a field which is the data element normally referenced within publication designs.

All records and fields to be treated as repeating data must be sequential. Data intended as repeating fields must either always appear a fixed number of times or have a separate field within the same record that specifies the number of times the element/group appears (the 'governor field').

Keyed data record delineation methods



carriage return / line feed

line feed only

Record descriptor word (RDW)

The RDW is a two-byte prefix that indicates the length of logical records within a stream. The pair can be either ‘Most Significant Byte’ (MSB) or ‘Least Significant Byte’ (LSB) ordered. MSB is commonly used on IBM operating systems. LSB is typical elsewhere. For RDW input DOC1 can also allow for an ‘eye-catcher’ string that is sometimes used for visual identification of records in a binary stream.

Note that you must not use the same key for records in different parts of the data structure even if they are otherwise identical. The key field must be unique in each instance, see “Identifying and defining fields” on page 251.

Byte and character measured data

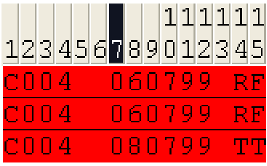
Byte measured – you can use this option to mark up *single byte input data*, for example, Latin based text where each character is represented by a single byte.

Character measured – this option is necessary when marking up double-byte based text such as, Chinese, Japanese and Korean data.

When working with character measured data the following restrictions apply.

- The encoding scheme is set for the entire data format. You cannot assign a different encoding scheme at record and field level.
- Binary field types are not supported.
- Packed field types are not supported.

Viewing keyed records in DOC1

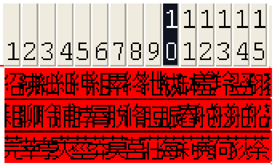


```
         111111
123456789012345
C004  060799 RF
C004  060799 RF
C004  080799 TT
```

Byte measured data

Each grid position in **byte measured** data is equivalent to a single character /byte.

Each grid position in **character measured** data is a single character, which in this case occupies two bytes.



```
         111111
123456789012345
```

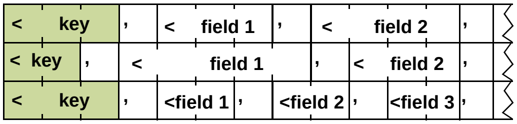
Character measured data

Delimited

This is a variation of the keyed record format in which fields are separated by one or more consistent text characters. This is sometimes known as a ‘comma-separated values’ or CSV format.

When working with delimited data field offsets and lengths are automatically identified. The key field must always be the first field in each record.

All fields in a delimited file can have variable length including the key field itself.



< key , < field 1 , < field 2 ,
< key , < field 1 , < field 2 ,
< key , <field 1 , <field 2 , <field 3 ,

This sample uses a single comma to delimit fields within records. The delimiter must always be the same one or more characters throughout the data.

Note that RDW prefixes are not supported for delimited input data – the key field must always appear first in each record. Records can be delineated using CR/LF or LF as with keyed record data.

Delimited is a text only format and binary field types are not supported. If field data needs to contain the actual delimiter character(s) the entire string must be enclosed in quote marks characters (defined in the data format). In the following example a single comma is defined as the separator and the double quote symbol as the quote mark:

```
001,Smith,"101, Princes Street",ED1 4DH
002,Debit,89.01,"Cash, charges apply"
```

As with the keyed record format, all records and fields to be treated as repeating data must be sequential.

XML

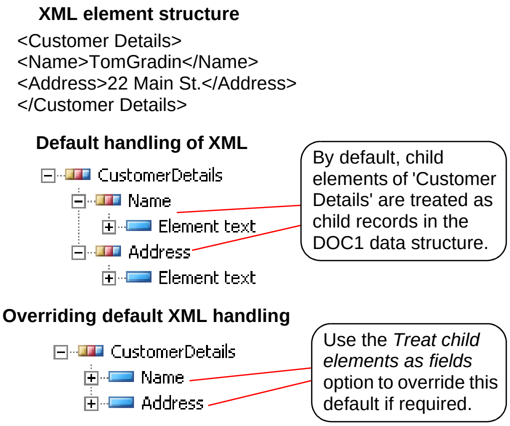
An Extensible Markup Language (XML) file used as input to DOC1 must have a document type definition (DTD) or schema (XSD) associated with it either within the XML or in an independent file.

Note that XML support is an optional extra and is only enabled to users who install both the DOC1 Work Center and Generate with the appropriate license keycodes.

Any 'well-formed' XML file should be acceptable as DOC1 input. The following XML features are not currently supported:

- Model groups
- XSL transformations
- Boolean data type

By default, DOC1 maps XML data items to its internal data structure by converting elements to records and attributes to fields. You can choose to treat nested XML records as either individual child records or as fields within their parent record.



XML element structure

```
<Customer Details>
<Name>TomGradin</Name>
<Address>22 Main St.</Address>
</Customer Details>
```

Default handling of XML

By default, child elements of 'Customer Details' are treated as child records in the DOC1 data structure.

Overriding default XML handling

Use the *Treat child elements as fields* option to override this default if required.

Working with sample data

Normally you will want to import your sample data directly into a data format. See "Creating a data format" on page 244 for details. This will result in the file being added to the Sample Data tab within the Data section of the DOC1 Work Center.

You can include additional sample data files in the repository as required and swap between files of the same type when working with the editors.

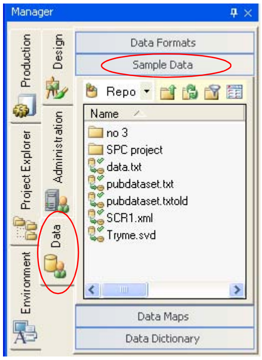
The Work Center allows you to browse the content of sample data files within the repository and to make minor adjustments to them if required.

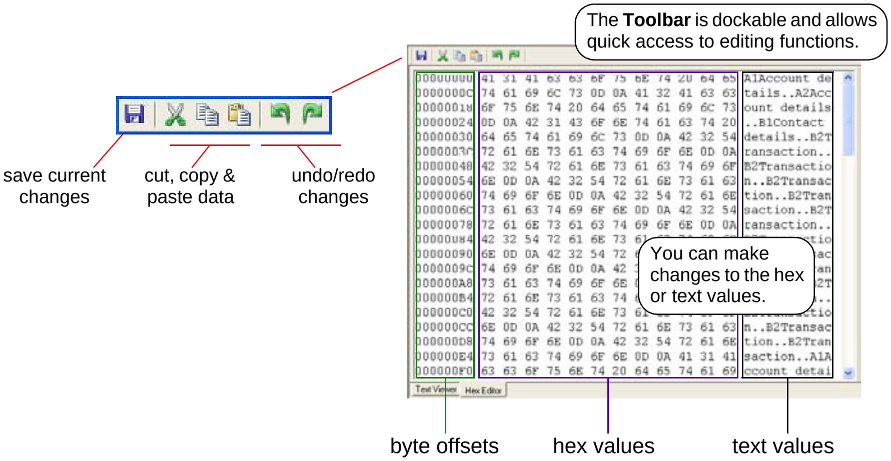
ⓘ CARE REQUIRED. IF YOU MAKE STRUCTURAL CHANGES TO A SAMPLE DATA FILE IT MAY NO LONGER BE SUITABLE FOR THE INTENDED DATA FORMAT.

You can also extract sample data files from the Work Center. This enables you to take a sample data file from the repository and save it to a location on your local directory.

▶ To browse or edit a sample data file:

1. In the Work Center Manager select the *Data* tab and then click the *Sample Data* bar.
2. Double-click the file you want to work with. This will be presented in the Sample Data Editor.
3. Use the *Text Viewer* tab to browse or switch to the *Hex Editor* page to make adjustments if required.




The **Toolbar** is dockable and allows quick access to editing functions.
save current changes
cut, copy & paste data
undo/redo changes
You can make changes to the hex or text values.
Text Viewer
Hex Editor
byte offsets
hex values
text values

```
00000000 41 31 41 63 63 6F 75 6E 74 20 64 65 A1Account de
0000000C 74 61 69 6C 73 0D 0A 41 32 41 63 63 tails..A2Acc
00000018 6F 75 6E 74 20 64 65 74 61 69 6C 73 ount details
00000024 0D 0A 42 31 43 6F 6E 74 61 63 74 20 ..B1Contact 
00000030 64 65 74 61 69 6C 73 0D 0A 42 32 54 details..B2T
0000003C 72 61 6E 73 61 63 74 69 6F 6E 0D 0A ransaction..
00000048 42 32 54 72 61 6E 73 61 63 74 69 6F B2Transactio
00000054 6E 0D 0A 42 32 54 72 61 6E 73 61 63 n..B2Transac
00000060 74 69 6F 6E 0D 0A 42 32 54 72 61 6E tion..B2Tran
0000006C 73 61 63 74 69 6F 6E 0D 0A 42 32 54 saction..B2T
00000078 72 61 6E 73 61 63 74 69 6F 6E 0D 0A ransaction..
00000084 42 32 54 72 61 6E 73 61
00000090 6E 0D 0A 42 32 54 72
0000009C 74 69 6F 6E 0D 0A 42
000000A8 73 61 63 74 69 6F 6E
000000B4 72 61 6E 73 61 63 74
000000C0 42 32 54 72 61 6E 73 61
000000CC 6E 0D 0A 42 32 54 72 61 6E 73 61 63 n..B2Transac
000000D8 74 69 6F 6E 0D 0A 42 32 54 72 61 6E tion..B2Tran
000000E4 73 61 63 74 69 6F 6E 0D 0A 41 31 41 saction..A1A
000000F0 63 63 6F 75 6E 74 20 64 65 74 61 69 ccount detai
```

▶ To extract a sample data file:

1. Select the *Data* tab and then click the *Sample Data* bar.
2. Highlight the file you want to extract and select *Tasks/Extract Sample Data*.
3. Browse to the folder where you want to extract the sample data and click *OK*.
4. Click *OK* in the *Extraction Log* dialog box and the procedure is complete.

Creating a data format

A data format is the DOC1 file that describes the type and structure of the input data file intended to be used with a particular job.

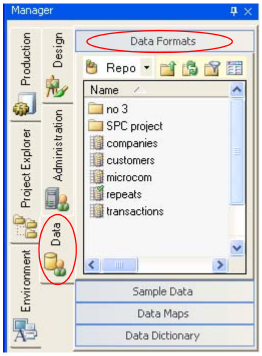
To create a data format you must import a sample of the data file that will be used in the production environment. This acts as the template on which the data format is built. The sample data file used for this purpose must contain examples of every type of data element that need to be referenced by the jobs that will use it.

The import process will automatically ascertain as much information as it can about the record and field structure of your input data. Once it is complete you can use the Data Format Editor to specify additional details about the structure. The amount of manual definition required will depend on the type of input data being used.

You can import a legacy data format from DOC1 Suite 4 into DOC1 Series 5. When used with a sample data file imported into the repository, the data format will be predefined in accordance to the definitions set out in the Data Format Editor in DOC1 Suite 4. After it has been imported it can then be manipulated like a data format created in the Work Center.

▶ **To import sample data into a new data format**:

1. In the Work Center Manager select the *Data* tab and then click the *Data Formats* bar.
2. Select *Tasks/Import Sample Data* from the menu.
3. Use the *Import Sample Data* dialog box to browse for the file to be imported and click *Open*.
4. In the *Sample Data File Properties* dialog box select the type of data being used and any related options.
5. Click *OK*. The Data Format Editor is invoked with the sample data file loaded ready for you to add specific definitions for records and fields within the structure. See “About the Data Format Editor” on page 247 for details.



▶ **To import a legacy data format file:**

1. Select the *Data* tab and then click the *Data Formats* bar.
2. From the menu select *Tasks/Import Legacy Data Format*.
3. Use the *Import* dialog box to browse to the required .ldf file and click *Open*.
4. As the data format is loading in the Data Format Editor another dialog box is displayed. Select the corresponding sample data file for the data format and click *Open*.

5. Select the *Define/Sample Data File Properties* menu option to set the default settings for the sample data and click *OK*. Proceed to define the data as described in “Working with keyed record and delimited data” on page 248.

> If you are going to import a legacy data format containing Ebcdic characters you must ensure you have selected *View/Undefined Data/As Ebcdic* before saving the LDF in the Data Format Editor in DOC1 Suite 4.

Import Sample Data File options

Data Type
Indicate the format of the sample data file being imported from the list of supported types. Refer to “Types of input data” on page 240 for detailed information.

Delineation
For keyed record or delimited data, use this option to indicate how the records within the file are delineated.
See “Working with keyed record and delimited data” on page 248 for details of supported data types and delineation methods.

Delimiter
For delimited data only. This option indicates the character(s) that are used to separate fields. Include spaces if these form part of the delimiter itself.

Quote
For delimited data use this option to indicate the quote mark character(s) used within the data. This will be used to identify the start and end of quoted string within the data. You may want to use quoted strings if your delimiter character(s) are also likely to appear as actual field data.

First record has column headings
For delimited data only. When selected the first record in the sample data is considered to be a column heading. Its values will be used to provide a default name for subsequent records and fields in the data format. The first record is ignored when the data is processed. Note that all instances of the ‘header’ record (i.e.with the same key) will also be ignored.

RDW includes prefix length
For keyed record data using RDW delineation. Check this option if the record length indicated in the RDW includes the two bytes of the RDW itself. If this is not checked the length of the RDW is assumed to be excluded from this value.

Eye catcher size
For keyed record data using RDW delineation. An eye catcher is a string that is sometimes used as an additional prefix to allow visual identification of individual records in a binary stream. If your data contains eye catchers, use this option to indicate the number of bytes they occupy in each record (this position is then ignored by DOC1 when processing the input data).

(continued)	**Import Sample Data File options**
Default encoding scheme	All data needs to be interpreted using a code page. System code pages included with DOC1 should be sufficient for most users, however, if you have imported a Custom code page that option will be enabled. This allows applications to understand the intended strings and values within the binary encoding that makes up the data itself. Use this option to identify the code page that will, by default, be assumed for interpreting the input data being imported. Note that parts of your data may use different encoding schemes (in particular, where text is based on multiple languages that use different code pages) so DOC1 allows you to override this default for each individual field if required.
Default Locale	This option allows you to specify a default locale to be used to determine regional delimiters within number and date fields. It can be overridden for individual field definitions if required.
Default Date Handling	See “Configuring date options”.
Default string options	This option enables you to have any spaces to the left or right of a data field removed when it is inserted into a document. This setting is applied to all string fields in the sample data, however, it can be overridden for specific fields using either the *Define Field* or the *Format* dialog box.
Empty Field Handling	When an empty data field is encountered in a publication at production time this option enables you to choose how to handle it. You can select to *Treat as Fatal Error* or *Replace With* a constant value. This setting is applied to all fields in the sample data, however, it can be overridden for specific fields using the *Define Field* dialog box or by using the *IsSet* and *IsnotSet* operators in a condition. For more details, see “Appendix B – Formula expressions” on page 414.

Configuring date options

Where your data contains date fields you may need to define how DOC1 should handle incomplete or invalid values. In particular, you may want to customize how two digit year values are interpreted.

▶ **To customize date options:** click the *Date Options* button on the *Sample Data File Properties* dialog box.

Date options

Two digit year conversion boundary

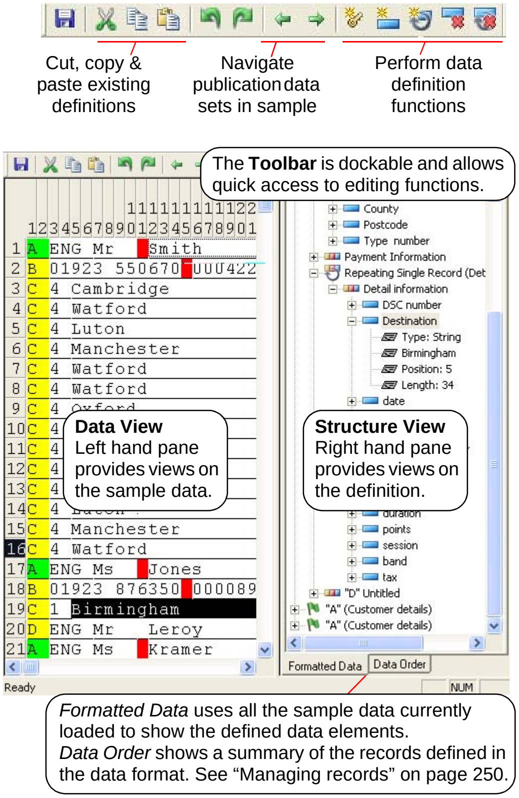
Date fields that have the year component stored as only two digits are always converted to a full four digit number when used in DOC1. Use this field to define the split between the 20th and 21st centuries for this conversion. The year is calculated to be in the range 19*bv* to 20(*bv*-1), where *bv* is the boundary value. For example:

A value of 75 gives a range 1975 – 2074.
A value of 0 gives a range is 1900 – 1999.
A value of 100 gives a range is 2000 – 2099.

About the Data Format Editor

The Data Format Editor (DFE) window always has two main panes. The left hand pane gives you a view on the sample data that was used to create the data format (or the latest version of the sample data to be loaded into the editor). The right hand pane shows you how DOC1 will interpret the data – i.e. the structure of records and fields into which the data is placed – when it is used with a production job.

The actual views available in these two panes will differ depending on what type of input data you are working with. The methods used to carry out the additional definition work required once the sample data has been imported will also vary. See “Working with keyed record and delimited data” on page 248.
and “Working with XML” on page 258 for details.



Cut, copy & paste existing definitions

Navigate publication data sets in sample

Perform data definition functions

The **Toolbar** is dockable and allows quick access to editing functions.

```
           1111111111122
  123456789012345678901
 1A ENG Mr    Smith
 2B 01923 550670 UUU422
 3C 4 Cambridge
 4C 4 Watford
 5C 4 Luton
 6C 4 Manchester
 7C 4 Watford
 8C 4 Watford
 9C 4 Oxford
10C 4
11C 4
12C 4
13C 4
14C 4
15C 4 Manchester
16C 4 Watford
17A ENG Ms    Jones
18B 01923 876350 000089
19C 1 Birmingham
20D ENG Mr    Leroy
21A ENG Ms    Kramer
```

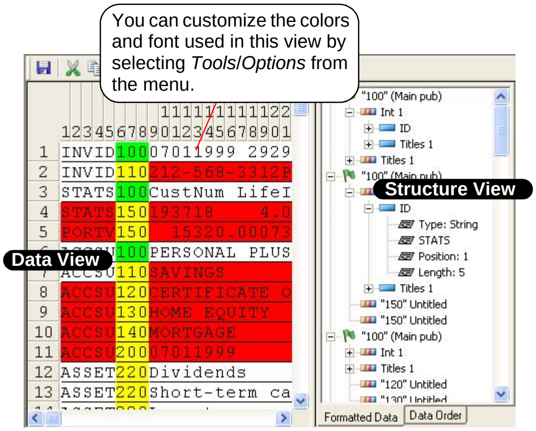
Data View
Left hand pane provides views on the sample data.

Structure View
Right hand pane provides views on the definition.

Formatted Data uses all the sample data currently loaded to show the defined data elements.
Data Order shows a summary of the records defined in the data format. See “Managing records” on page 250.

Working with keyed record and delimited data

With keyed record or delimited data loaded, the data view shows you the content of you sample data in a grid format. This is used to mark specific parts of the data when defining the extent of elements. The structure view shows you the records and fields you have defined and allows you to view and control their properties.



You can customize the colors and font used in this view by selecting *Tools/Options* from the menu.

```
          11111111122
 123456789012345678901
 1 INVID10007011999 2929
 2 INVID110212-568-3312P
 3 STATS100CustNum LifeI
 4 STATS150193718     4.0
 5 PORTV150   15320.00073
 6 ACCSU100PERSONAL PLUS
 7 ACCSU110SAVINGS
 8 ACCSU120CERTIFICATE O
 9 ACCSU130HOME EQUITY
10 ACCSU140MORTGAGE
11 ACCSU20007011999
12 ASSET220Dividends
13 ASSET220Short-term ca
```

```
"100" (Main pub)
    Int 1
        ID
        Titles 1
    Titles 1
"100" (Main pub)
    
        ID
            Type: String
            STATS
            Position: 1
            Length: 5
        Titles 1
    "150" Untitled
    "150" Untitled
"100" (Main pub)
    Int 1
    Titles 1
    "120" Untitled
    "130" Untitled
```

The Data Format Editor with keyed record data

These data types essentially require you to use the same features in the DFE although with delimited data much of the information about the structure of the data can be assumed.

For keyed record data you will need to:

- identify the key field
- define the location of all individual fields that need to be referenced by a DOC1 job
- define controls for fields that have varying size.

For both keyed record and delimited data you will need to:

- indicate the record types that mark the start of a publication
- mark records that must always appear in a publication data set as *mandatory*
- define a reference name and data type of all fields that need to be referenced by a DOC1 job
- identify the record and field types that will appear iteratively – i.e. repeating data.

Defining the key field

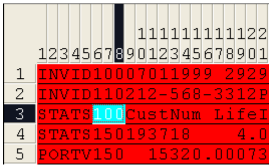
Individual record types within keyed record data are identified according to the contents of the key field. The contents of this field must uniquely identify the record type and must be the same in each record of the same type.

For delimited data the key field is automatically assumed to be the first field in each record and cannot be changed but it can have varying size.

For keyed record data the key field must appear at the same position within every record in the input data and can be 1-64 characters in length. The field length cannot vary in different records and its content must be padded with spaces if necessary.

When working with keyed record data no other data definitions can take place until the key field is defined.

▶ **To define or redefine the key field**:



```
              1111111111122
     123456789012345678901
1    INVID10007011999 2929
2    INVID110212-568-3312P
3    STATS100CustNum LifeI
4    STATS150193718     4.0
5    PORTV150  15320.00073
```

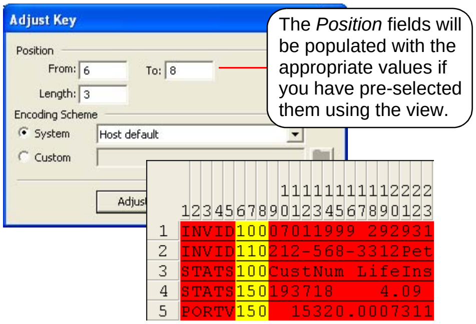
1. If required, select the columns to act as the key.

   In the data view drag the mouse over the required positions in any record.

2. Select *Define/Key* from the menu or click the define key button on the toolbar and the *Define Key* dialog box is displayed.

3. Confirm or enter the key field offset and length and if necessary select a different encoding scheme to that chosen as the default scheme for the data format. Note that the option to change the encoding scheme is only available when working with keyed record (byte measured) input data.

4. Click *Define* to establish the key field. The appropriate columns are highlighted.



Adjust Key

Position

From: 6 To: 8

Length: 3

Encoding Scheme

● System Host default

○ Custom

Adjus

The *Position* fields will be populated with the appropriate values if you have pre-selected them using the view.

```
              11111111112222
     12345678901234567890123
1    INVID10007011999 292931
2    INVID110212-568-3312Pet
3    STATS100CustNum LifeIns
4    STATS150193718     4.09
5    PORTV150  15320.0007311
```

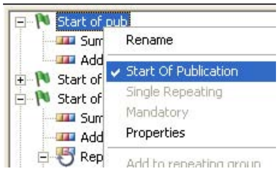
Working with records

DOC1 is automatically aware of the various record types within the sample data once the key field has been defined. Records assume a default label from the contents of the key field. You may often want to assign more meaningful names to the record types however.

Additionally, one or more records that indicate the start of a new publication data set will need to be identified.

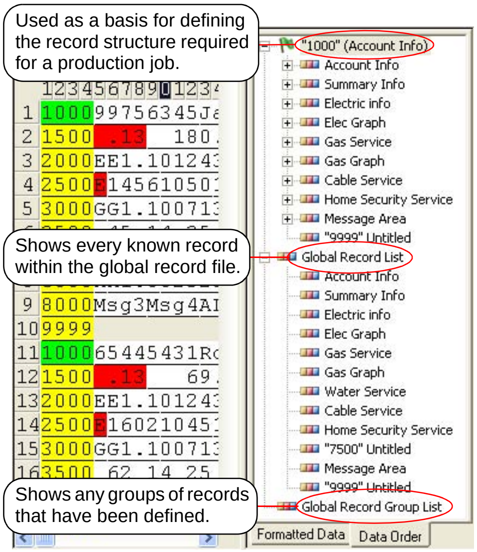
▶ **To define a start of publication record:** right-click on it in the structure view and select *Start of Publication* from the shortcut menu.

▶ **To rename a record:** right-click on it in the structure view and select *Rename* from the shortcut menu. Then overtype the name within the view itself.



Managing records

The *Data Order* tab allows you to manage the record structure used in publications, for example, re-create the record order, exclude records or group them together.



Used as a basis for defining the record structure required for a production job.

Shows every known record within the global record file.

Shows any groups of records that have been defined.

Recreating record order

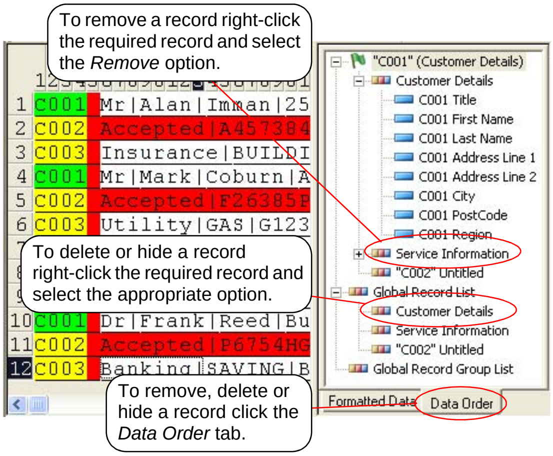
If necessary you can remove the data hierarchy of a publication data set that was previously defined. This allows you to redefine the data structure for a production job.

▶ **To recreate the record order:** in the structure view select the *Data Order* tab. Right-click on the appropriate global record file and select *Recreate record order* from the shortcut menu.
Use this option with care as it flattens the file and removes all previously defined hierarchy in the formatted data, such as repeating groups.

Excluding Records

You can *Remove* a record from the formatting but if the record is used in the current data it will be retained for future use.

The *Delete* option removes the record format in every data publication data set in which the record is used.
Records that used it will be shown as unformatted.



To remove a record right-click the required record and select the *Remove* option.

To delete or hide a record right-click the required record and select the appropriate option.

To remove, delete or hide a record click the *Data Order* tab.

If you *Hide* the record it will not be processed but the formatting remains. To reverse the hide status repeat the process and the tick is removed, reactivating the record.

ⓘ After removing or deleting records the screen is refreshed and a new record format is created with the name "Untitled" for each record in the current data that is unformatted.

Grouping records

If you want to use the same record at different locations within the data format, then you must use different key numbers for each record. However, you can group these records together, which will enable you to maintain them as one, i.e. changes made to any of the fields will be applied to all records in that group.

▶ **To group records together:** in structure view select the *Data Order* tab. Highlight *Global Record Group List* and select *Create Record Group* from the shortcut menu. In the *Record Group* dialog box, select the record from the record list that you want to use as the template, and *Add* it to the group. Repeat for all other records that you want in the group.

▶ **To modify a record group:** in structure view select the *Data Order* tab and expand the *Global Record Group List*. Right-click the record group and select *Edit Record Group* to display the *Record Group* dialog box and make the required changes.

▶ **To delete a record group:** in structure view select the *Data Order* tab and expand the *Global Record Group List*. Right-click the record and select *Delete Record Group*.

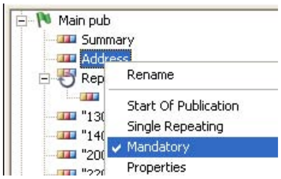
Mandatory records

By default, records other than those flagged as *Start of Publication* do not have to appear in every publication data set. Provided fields within the record are used conditionally within document designs they will simply be ignored when they are not present.

However, it may be important to your document design to make sure that certain records are always present before attempting to process a publication. Where this is the case you can flag such records as being *Mandatory*.

If a mandatory record does not appear in a publication data set where it was expected DOC1 Generate will follow the *on error with publication* setting which is defined when publishing.

▶ **To define a record as being mandatory:** right-click on it in the structure view and select *Mandatory* from the shortcut menu.



Identifying and defining fields

You will need to define each field within the input data that you want to use as part of a document design.

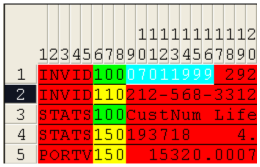
- For keyed record data you will need to indicate the position and length of fields. (For delimited data this information can be assumed from the file format itself.)
- For both keyed record and delimited data you will need to give each field a reference name and specify the type of data it will provide.

Most keyed record fields will have a consistent offset and length within records. Variable length fields are supported providing they have a marker that indicates the end of the field contents. This marker must consist of a fixed length constant of one or more characters which are appended to the field contents proper whenever it appears.

▶ **To define a field**:

1. If you are working with keyed record data first identify the field position. In the data view drag the mouse over the required positions in any example of the record in which the field appears.

   For delimited data or a field that has already been defined simply click within any example of the appropriate field.



```
            1111111112
   12345678901234567890
 1 INVID10007011999 292
 2 INVID110212-568-3312
 3 STATS100CustNum Life
 4 STATS150193718    4.
 5 PORTV150  15320.0007
```

2. Select *Define/Field* from the menu or click the define field button on the toolbar.
3. In the *Define Field* dialog box you must specify the type of data the field contains and any related settings (see below).
4. Click *Define* to add the field definition to the data format.

Define Field dialog box

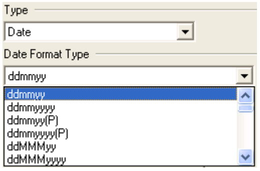
Position — For keyed record data use these settings to adjust the field offset and length if required. Be aware of the effects changes might have on other field definitions if you make changes here.

Type — Identifies the type of data the field contains.

String — Can be either text data or RTF data with basic text formatting wi.th a maximum size of 32k characters. With string types you can choose to override the default string options by selecting *Trim Left* and *Trim Right*.

Date — A calendar date. If this type is selected you must use the *Date Format Type* field to identify the sequence and format of the date components. Note that any date separator characters within the input data are automatically allowed for and are not shown in the list. Month values provided as text are interpreted according to the settings in the default or field specific *Locale*. For instance, if the selected locale includes the full name of the first month of the year as 'Janvier' the month component in the field must match this to be correctly interpreted. The meaning of the various date component options are:

dd – day of the month
ddd – Julian day of the year (e.g. 032 = 1 February)
mm – month as a two-digit code (e.g. 01 = January)
MMM – month as a three character text abbreviation
MMMM – month as a full case sensitive string
yy – year as a two digit abbreviation
yyyy – year as a full four digit number
(P) – the date is stored as packed decimal.



Type
Date
Date Format Type
ddmmyy
ddmmyy
ddmmyyyy
ddmmyy(P)
ddmmyyyy(P)
ddMMMyy
ddMMMyyyy

(continued) **Define Field dialog box**

Click *Options* if you want to override the default settings for interpreting date components for the field.

See “Working with keyed record and delimited data” on page 248 for details of date interpretation settings. See “Creating and customizing locales” on page 184 for information about locales.

Number A signed or unsigned number with an optional floating point. Values can have a maximum of 24 digits before the decimal place and 8 after. Where used, the decimal point is real – i.e. a byte containing a decimal point character is stored as part of the number. The character used as the decimal point is interpreted according to the settings in the default or field specific *Locale*.

Decimal A signed or unsigned number with optional implied decimal places. The use of this option invokes the *Precision* entry field for the position of the decimal point to be defined. Maximum value is the same as type *Number*.
Note that the binary and packed field options are not available when working with character measured keyed data.

Native Binary A binary encoded number where the storage type (LSB or MSB as in the options above) is assumed from the operating system on which DOC1 Generate processes a job. Decimal places are implied – use the *Precision* field as with type *Decimal*

LSB Binary A binary encoded number where the most significant byte is the second in a two-byte pair (typical of applications from Windows and Unix systems). Decimal places are implied – use the *Precision* field as with type *Decimal*.

MSB Binary A binary encoded number where the most significant byte is the first in a two-byte pair (typical of applications using IBM operating systems). Decimal places are implied – use the *Precision* field as with type *Decimal*.

Packed Decimal A signed number represented by the hex values of the field bytes, two digits per byte. The second nibble of the final byte indicates the sign: hex 'D' indicates a negative value, all others a positive. Thus the hex string X'348961' represents the value +34896. Decimal places are implied – use the *Precision* field as with type *Decimal*.

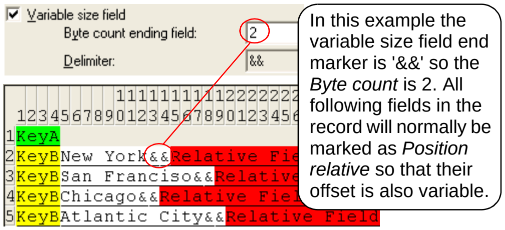
Sign Overstruck As for *Packed Decimal* except that the first nibble of the final byte indicates the sign.

Position relative to previous field Fields that follow data structures of variable length within a record – i.e. fields that appear after variable size fields or repeating fields – should have this flag turned on. Under normal circumstances this option should be automatically checked where appropriate.

(continued) Define Field dialog box

Variable size field

Checking this option indicates the field can occupy a varying number of bytes. In order to use this format the field must always be terminated by the same delimiter characters. The length of the delimiter is specified as the *Byte count ending field* setting



☑ Variable size field
Byte count ending field: 2
Delimiter: &&

```
          11111111112222222
 12345678901234567890123456
1KeyA
2KeyBNew York&&Relative Fie
3KeyBSan Franciso&&Relative
4KeyBChicago&&Relative Fiel
5KeyBAtlantic City&&Relative Field
```

In this example the variable size field end marker is '&&' so the *Byte count* is 2. All following fields in the record will normally be marked as *Position relative* so that their offset is also variable.

Encoding scheme

This option allows you to specify the system code page to be used when interpreting the data within the field – i.e. the code page for which the field data was intended to be used with. A default code page is assigned when the data format is first created by importing sample data and you should not need to change this setting unless parts of your data are based on different encoding schemes (in particular, where text is based on multiple languages that use different code pages). *System* code pages included with DOC1 should be sufficient for most users. However if you have imported a *Custom* code page that option will be enabled.
Note that the Encoding scheme options will be disabled for comma separated data (CSV).

> ⓘ IT IS IMPORTANT TO NOTE THAT IF YOUR INPUT DATA IS CHARACTER MEASURED, THE OPTION TO DEFINE AN ENCODING SCHEME AT FIELD LEVEL IS NOT AVAILABLE. A SINGLE ENCODING SCHEME IS SET FOR THE ENTIRE DATA FORMAT WHEN WORKING WITH THIS TYPE OF INPUT DATA.

Locale

This option allows you to override the default locale specified for the data format and assign a specific locale to be used when interpreting the field. It is used to determine regional delimiters within number and date fields.

Empty Field Handling

When an empty data field is encountered in a publication at production time this option enables you to choose how to handle it. You can select to *Treat as Fatal Error* or *Replace With* a constant value. This setting overrides the default settings for the data format. You can also use the *IsSet* and *IsnotSet* operators in a condition to set the handling options. For more details, see “Appendix B – Formula expressions” on page 414.

Editing field definitions

When working with keyed record data you can copy or move the definition assigned to an existing field to create a new field definition at a different location.

You can redefine existing field definitions to change attributes or amend the defined position of the field. When moving field definitions, note that if the new location occupies any positions already defined for other fields these will automatically be undefined. It is often cleaner to undefine the affected fields specifically before redefining their positions.

You may also want to shift a group of existing field definitions along the record offset range particularly if you are editing a data format to take account of additional data that has been added to a record.

▶ **To edit a field definition:** double-click on it in either the data or structure view
– or –
with the field highlighted select *Define/Field* from the menu or click the toolbar button
– or –
right-click on it in the structure view and select *Properties* from the shortcut menu.

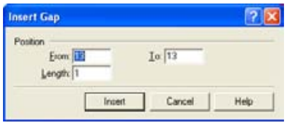
▶ **To undefine a field definition:** with the field highlighted select *Define/Remove Field* from the menu or click the toolbar button.

▶ **To copy or move a field definition**:

1. Click on the field in either the data or structure view and select *Copy* or *Cut* from the Edit menu
   – or –
   right-click on the field and select the required option from the shortcut menu.
2. Identify the start position for the new field by clicking in an undefined area in the data view.
3. Select *Edit/Paste* from the menu
   – or –
   right-click in the data view where the definition is to be inserted and *Paste* from the shortcut menu.
4. Make any necessary adjustments in the *Define Field* dialog box and click *Define* to complete the new definition.

▶ **To shift field definitions**:

1. In the data view click on a field adjacent to where a gap is to be inserted or removed.
2. Select *Insert Gap* or *Close Gap* from the Define menu.
3. Use the dialog box to specify the positions to be inserted or removed from the record definition.



From: 13
To: 13
Length: 1

Defining repeating data structures

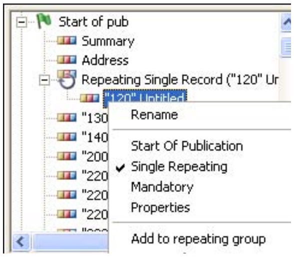
Repeating data are iterations of data elements of the same type within the input file. These typically contain transactional information such as the details of an itemized phone bill or a list of purchases on a credit card.

For keyed record data, DOC1 supports two kinds of repeating data structure: repeating records and repeating fields. Each type can either be an iteration of a single element or of a group of elements – e.g. a range of several elements which appears iteratively as a group. Groups of repeating elements must appear sequentially in the input data.

It is possible to nest repeating records/groups and to have repeating fields/groups within repeating records. Nested repeating fields are not supported.

Repeating records

Repeating records are specified using properties of the record objects in the structure view.

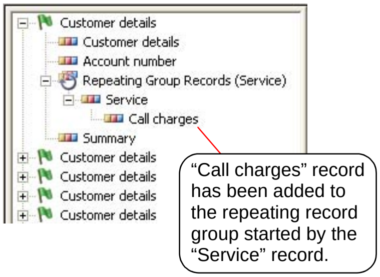


▶ To define or undefine a single repeating record: right-click on the appropriate record in the structure view and select *Single Repeating* from the short cut menu.

▶ To define a repeating record group:

1. Identify the first record that makes up the repeating record group.
2. Right-click on this record in the structure view and select *Single Repeating* from the short cut menu.
3. Right-click on a subsequent record that is to be part of the group and select *Add to repeating group* from the shortcut menu.
4. Repeat this for each of the records that make up the repeating group.

▶ To remove a record from a repeating group: right-click on it in the structure view and select *Remove from repeating group* on the shortcut menu.



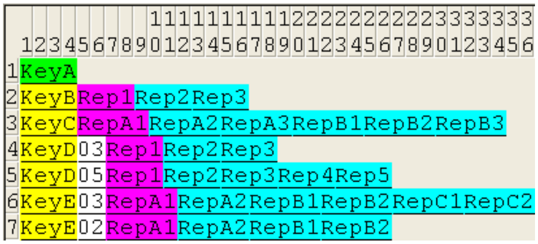
“Call charges” record has been added to the repeating record group started by the “Service” record.

Repeating fields

Unlike repeating records, fields cannot be iterated an unlimited number of times. DOC1 needs to be told how many times the field or group of fields are repeated each time it appears in the input data.

If the data is always repeated the same number of times you can specify the *fixed count* setting.

Alternatively, you can dynamically indicate the number of repeats by including a *governor field* within the input data. A Governor field must contain an integer value that indicates the number of times the group is repeated in the current record. It must appear in the same record and before the repeating fields to which it relates, but it need not be contiguous with the repeating data itself.



```
         111111111122222222223333333
123456789012345678901234567890123456
1KeyA
2KeyBRep1Rep2Rep3
3KeyCRepA1RepA2RepA3RepB1RepB2RepB3
4KeyD03Rep1Rep2Rep3
5KeyD05Rep1Rep2Rep3Rep4Rep5
6KeyE03RepA1RepA2RepB1RepB2RepC1RepC2
7KeyE02RepA1RepA2RepB1RepB2
```

KeyB: single field repeated a fixed number of times (3)

KeyC: group of 3 fields repeated a fixed number of times (2)

KeyD: single field with variable repeats specified in governor field (white)

KeyE: group of 2 fields with variable repeats specified in governor field (white).

▶ To define a repeating field (single or group):

1. All the data elements to be included in the repeat must already be defined as individual fields. See “Identifying and defining fields” on page 251 for details.

2. In the structure or data view click on the field that is to be the first (or only) in the repeating field group

> NOTE THAT ALL FIELDS OTHER THAN THE FIRST IN THE GROUP MUST BE DEFINED AS *RELATIVELY POSITIONED*. IF YOU DID NOT SPECIFY THIS WHEN DEFINING THE FIELDS YOU WILL BE ASKED TO CONFIRM THAT THIS SETTING SHOULD AUTOMATICALLY BE APPLIED.

3. Select *Define/Repeating Field* from the menu
   – or –
   click the toolbar button.

4. In the *Define Repeating Field* dialog box adjust the value in *Number of fields* if more than one field is to be included in the group.

5. **For a fixed number of repeats** click the *Fixed Count* radio button and enter the number of times the group is repeated in the application data in the *Number of repeats* entry field.

   ● Fixed Count 1

   For a variable number of repeats click the *Governor Field* radio button and select from the list of fields in the record (only those fields prior to the field being repeated are listed).

   ● Governor Field

   Fld Key Definition Field
   Fld Gov

6. Select *Define* to establish the repeating field group.

▶ To undefine or adjust a repeating field definition:

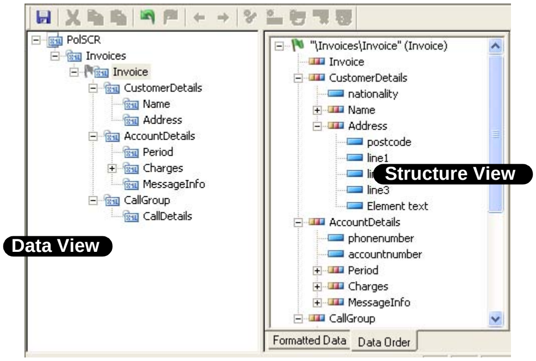
1. Highlight the first field of the group.
2. From the *Define* menu select *Remove Repeating Field* from the menu
   – or –
   click the *Remove Repeating Field* toolbar button.
3. Recreate the repeating field with new settings if required.

> ⓘ You cannot undefine a repeating field group if it has existing field definitions following it in the record (these will normally be marked as *relatively positioned* and therefore rely on the repeating field for their offset information). Undefine the subsequent field definitions or unset their relative positioning flag before un-defining the repeating field.

Working with XML

XML files used as input to DOC1 must have an associated document type definition (DTD) or schema (XSD) either within the XML itself or in an independent file. Where these are stored externally, the sample data import task will attempt to find the DTD or XSL file at the location specified within the XML or at the same location as the XML file if no path is specified.

XML is self-describing so the sample data import task can automatically determine the data elements it contains including any repeating data structures. “Creating a data format” on page 244 describes how this task is run. You will still need to use the Data Format Editor to indicate the record types that mark the start of publication data sets. In some circumstances you may also need to redefine some of the data types that have been assumed for individual fields.



The Data Format Editor with XML data

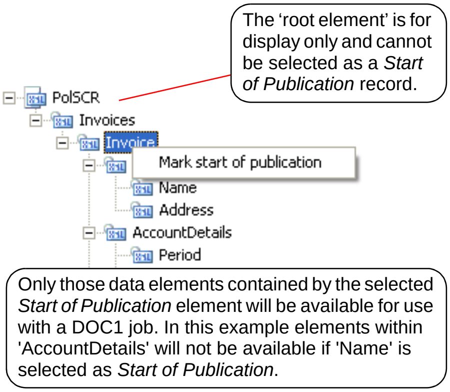
With XML data loaded, the data view shows you the data elements defined in the DTD or XML. These equate to records in the DOC1 data model. One or more of these records need to be marked as *Start of Publication*.

> ⓘ See “Data structure” on page 238 for more information about the DOC1 data model and how publication data sets are identified.

Once this has been done DOC1 will evaluate all the elements within the appropriate structures of the XML file itself. In the structure view individual data fields will be created for each identifiable data element.

By default, when XML data is evaluated elements are automatically treated as records and their attributes and text are treated as fields of those records. You may choose to override this default setting and have child XML elements treated as fields within their parent record instead.

▶ **To define a start of publication record:** right-click on it in the data view and select *Start of Publication* from the shortcut menu.



The ‘root element’ is for display only and cannot be selected as a *Start of Publication* record.

Only those data elements contained by the selected *Start of Publication* element will be available for use with a DOC1 job. In this example elements within 'AccountDetails' will not be available if 'Name' is selected as *Start of Publication*.

▶ **To change the data type associated with an element:** select the *Formatted Data* tab. Right-click the data field you want to change and select *Properties* from the shortcut menu. Choose the required data type from the *Type* list.

▶ **To override the default handling of XML elements/records:** double-click the record in the Structured View and the *Advanced Record Properties* dialog box is displayed. Select the *Treat child elements as fields* option and click *Define*.

Exporting to other objects

If required you can export the definitions within a data format into the other main data related DOC1 objects, a data dictionary and a data map.

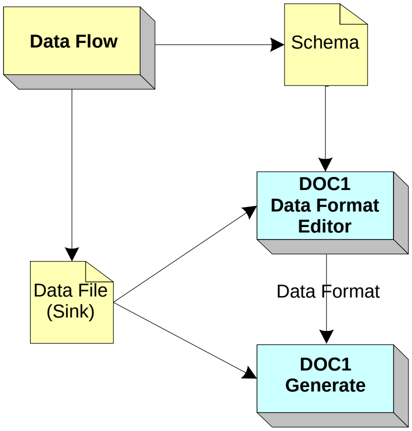
A data dictionary provides a list of data record and data field aliases that can be referenced when creating publication designs. A data map is used to link the aliases used by a publication to the elements within a data format and must be complete before it can be published.

See “Data mapping” on page 261 and “Working with data dictionaries” on page 267 for further information.

You may want to use this feature if you have a static input data environment and want to be confident that the aliases used in publication designs are represented in the actual input data file that will be available in the production environment.

▶ **To export a data dictionary and data map:** select the Data tab and then click the Data Formats bar. Highlight the data format that you want to use for the data map from the navigator. Select *Tasks/Generate Dictionary* to display the Data Dictionary and Data Map Wizard. The wizard will guide you through the generation process.

Importing data formats from Data Flow™



The Data Flow data integration solution allows you to select and join data from multiple sources into a single data feed.

When working with keyed record or delimited data, you have the option in Data Flow to create a schema describing the structure of the data file (Sink). This schema can be associated with your input data in the Data Format Editor to automatically define the record and field structure of your input data.

▶ **To mark up data automatically using a Data Flow schema:**

1. Create a new data format using your data from Data Flow, see to “Creating a data format” on page 244 for details.
2. From the *Define* menu select *Import Data Format Schema*
   – or –
   click the *Import Data Format Schema* toolbar button.
3. Locate and load the appropriate schema.
4. Assign a default locale to determine regional delimiters within number and date fields.

In addition to the import process, you can optionally export a schema from marked up data in the Data Format Editor, which can be used outside DOC1, if required.

▶ **To export an XML schema from marked up data in the Data Format Editor:** from the *Define* menu select *Export Data Format Schema*
– or –
click the *Export Data Format Schema* toolbar button.

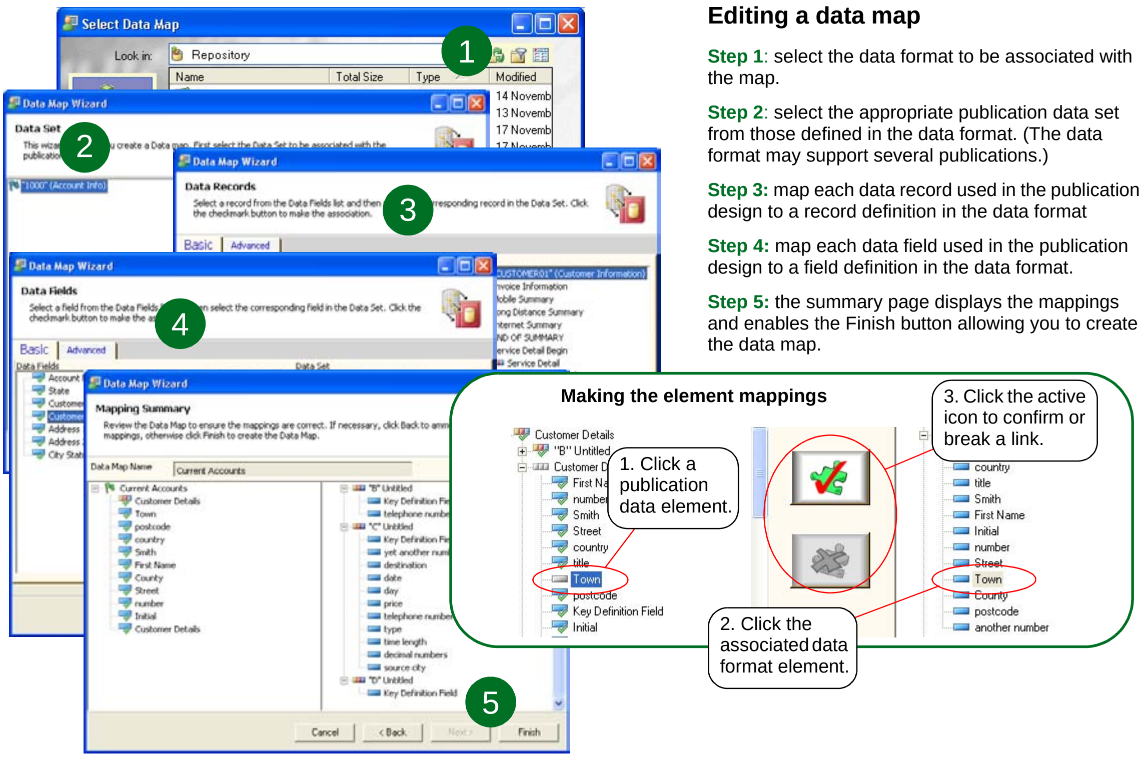
Data mapping

When you are designing publications in the DOC1 Work Center you work with indirect references to fields and records. Before you can publish a design all such aliases must be linked with actual data elements that are defined in the data format to be used with the publication in the production environment. These cross-references are defined and maintained in a data map object.

If you do not already have a suitable data map for a publication you will need to create one before you can complete the publishing task. You can do this as part of the publishing task itself.

REFER TO "RUNNING A PUBLISHING TASK" ON PAGE 336 FOR FURTHER INFORMATION.

Alternatively you can create a data map in advance and assign it to a publication. You will need to do this if you want to use a sample data file to show examples of 'real' data within a design before it is published.


Select Data Map
Look in: Repository
Name
Total Size
Type
Modified
14 Novemb
13 Novemb
17 Novemb
1
Data Map Wizard
Data Set
2
"1000" (Account Info)
Data Map Wizard
Data Records
Select a record from the Data Fields list and then ... corresponding record in the Data Set. Click the checkmark button to make the association.
3
Basic
Advanced
Data Map Wizard
Data Fields
Select a field from the Data Fields ... en select the corresponding field in the Data Set. Click the checkmark button to make the as...
4
Data Fields
Data Set
Data Map Wizard
Mapping Summary
Review the Data Map to ensure the mappings are correct. If necessary, click Back to am... mappings, otherwise click Finish to create the Data Map.
Data Map Name
Current Accounts
Cancel
< Back
Next >
Finish
5
Customer Details
Town
postcode
country
title
Smith
First Name
Initial
number
Street
County
Key Definition Field
another number

Editing a data map

Step 1: select the data format to be associated with the map.

Step 2: select the appropriate publication data set from those defined in the data format. (The data format may support several publications.)

Step 3: map each data record used in the publication design to a record definition in the data format

Step 4: map each data field used in the publication design to a field definition in the data format.

Step 5: the summary page displays the mappings and enables the Finish button allowing you to create the data map.

Making the element mappings

1. Click a publication data element.

2. Click the associated data format element.

3. Click the active icon to confirm or break a link.

You can specifically create a data map object and associate it with a data format and a publication. You can then add individual links between the data elements defined in each.

You can also export both a data map and a data dictionary from a data format. This will allow document designers to work with a list of known data elements and avoids the need to link individual elements manually.

Refer to "Exporting to other objects" on page 259 for further information.

Data maps can be edited using either the data mapping wizard or data map editor. The wizard allows you to define the required links using models of the data aliases used within a publication and the data elements defined in a data format.

Generating a dictionary and data map

The *Generate Dictionary* option automatically creates a dictionary and data map from a data format. The data elements in the data format are replicated as records and fields in the new dictionary with the mappings already completed in the new data map.

The Data Dictionary and Data Map Wizard can be used to create and update dictionaries and maps. It enables you to select which publication data set you want to use and what names you want to use for the new objects.

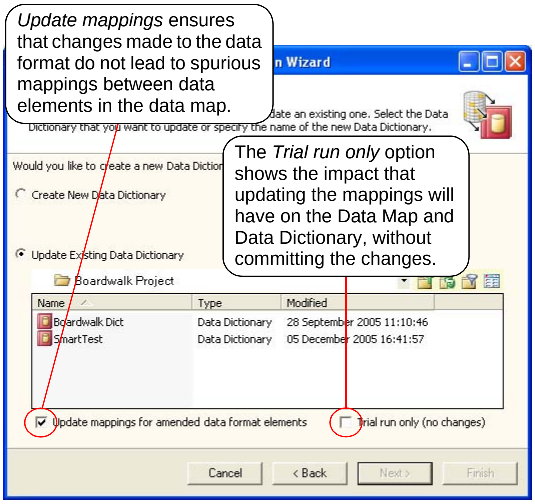
When updating a data dictionary and data map, additional options for handling changes to the data format are available. If changes are made to the data format, such as changing the data type of an element from a number to a decimal, you can use the *Update mappings for amended data format elements* option to rationalize the data mappings and ensure that spurious mappings between data fields and data elements do not occur.

Using the *update mappings* option every time you make a change to the data format will keep the data map from having multiple data fields mapped to the same data format element.

If you have a data dictionary and data map which have not made use of the *update mappings* option they may contain any number of extra data fields. As this may have been unavoidable in previous versions of DOC1, using the *update mappings* option on a legacy dictionary and data map will attempt to remove multiple data fields to data element mappings.

▶ **To generate a data dictionary and data map:** select the Data tab and then click the Data Formats bar. Highlight the data format that you want to use for the data map from the navigator. Select *Tasks/Generate Dictionary* to display the Data Dictionary and Data Map Wizard. The wizard will guide you through the generation process.

▶ **To update a data dictionary and data map:** select the data format used for the dictionary and data map to be updated and select *Tasks/Generate Dictionary*. After selecting the *Publication Data Set*, select the *update* options where applicable and the wizard will guide you through the update process. When it is complete a dialog box is displayed showing which mappings have been affected.



Update mappings ensures that changes made to the data format do not lead to spurious mappings between data elements in the data map.

The *Trial run only* option shows the impact that updating the mappings will have on the Data Map and Data Dictionary, without committing the changes.

n Wizard

date an existing one. Select the Data Dictionary that you want to update or specify the name of the new Data Dictionary.

Would you like to create a new Data Diction

○ Create New Data Dictionary

● Update Existing Data Dictionary

Boardwalk Project

Name	Type	Modified
Boardwalk Dict	Data Dictionary	28 September 2005 11:10:46
SmartTest	Data Dictionary	05 December 2005 16:41:57

☑ Update mappings for amended data format elements

□ Trial run only (no changes)

Data Mapping Wizard

Data maps can be created outside the Data tab using the Data Mapping Wizard. The wizard is used to map the publication data set from a data format to the data fields used in a publication. The Data Mapping Wizard can be invoked from the Publish Wizard, Production Job view or the *Publication* dialog box.

The Data Mapping Wizard provides a *Basic* and an *Advanced* view. The *Basic* view simply lists the publication data elements and enables them to be associated with a data format. The *Advanced* view provides the ability to search the lists. The same options are provided in the Data Map Editor, see “Editing a data map” on page 265 for more details.

▶ **To map a publication’s data fields:** invoke the Data Mapping Wizard by doing one of the following:

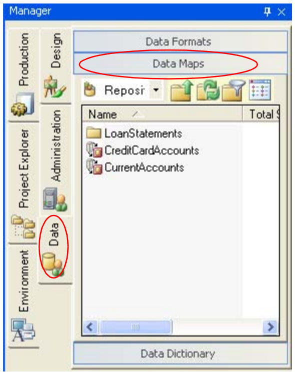
- in the Document Editor click *View/Publication* and double-click the publication in the logic map to display the *Publication* dialog box. Click on the *Data Map* button and select a data format.

- in the Design tab select the Publication bar. Select a publication from the navigator and click *Tasks/Publish…* to display the Publish wizard. Select the publication in the *Publications* list and click the *Assign Data Map* button.

- in the Production tab open a production job from the navigator. Select a publication from the *Publications* list and click the *Assign Data Map* button.

Creating a data map

Data maps can be created and edited from the Data Mapping Wizard when configuring a publishing job, however, they can also be created from the *Data* tab. From the *Data* tab data maps can be created independently of a publication, either using the *File/New Data Map* option or by using the *Generate Dictionary* feature.

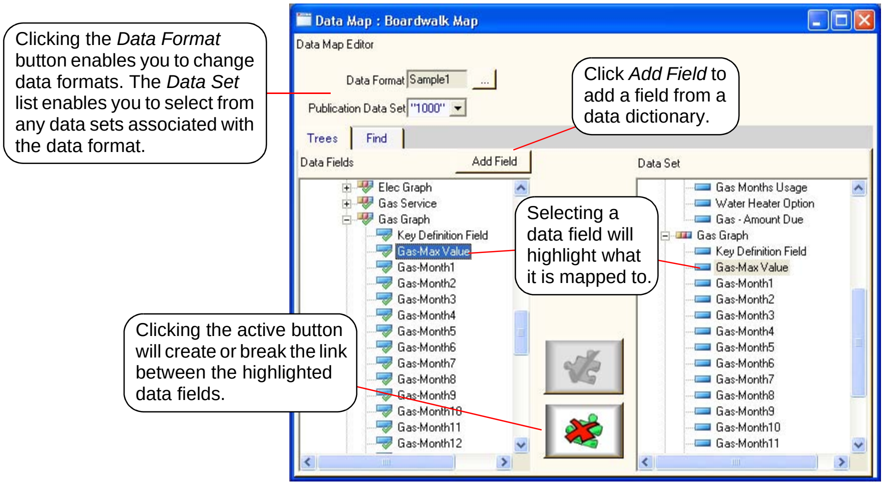


▶ To create a data map:

1. In the Work Center Manager select the *Data* tab and then click the *Data Map* bar.
2. Select *File/New Data Map* or right-click in the navigator and click *New Data Map*. A new data map object is displayed. Enter a name and press *Return*. The Data Map Editor is displayed.
3. In the *Tree* view click the *Data Format* browse button ... to select the data format to use with the data map.
4. From the *Publication Data Set* list, select the data set you want to use from the data format.
5. Click *Add Field* to select a record or field from a data dictionary you want to map to a record or field in the data set.

   Repeat the step for each record or field you want to include in the data map.
6. Select a field in the *Data Fields* list and select a corresponding field in the *Data Set* list. Click the *Check* button to map the fields.

ⓘ FOR MORE INFORMATION ON DATA DICTIONARIES, SEE “WORKING WITH DATA DICTIONARIES” ON PAGE 267.



Clicking the *Data Format* button enables you to change data formats. The *Data Set* list enables you to select from any data sets associated with the data format.

Click *Add Field* to add a field from a data dictionary.

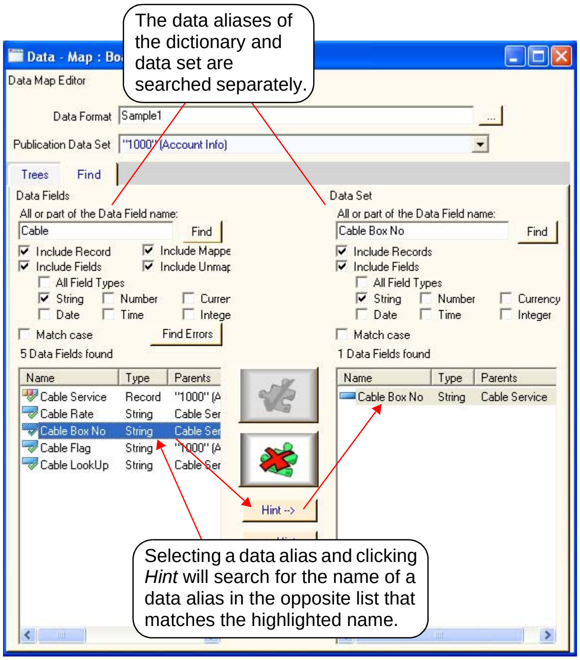
Selecting a data field will highlight what it is mapped to.

Clicking the active button will create or break the link between the highlighted data fields.

Editing a data map

Once a data map has been created you can change the data element mappings, as well as the data format used by the data map. These changes can be made in either the *Tree* or *Find* view.

The *Find* view enables you to search for data elements defined in either the data dictionary or the data format associated with the data map. This is useful when the data map contains a large number of elements. As well as being able to focus your search on specific types of records and fields, the *Hint* option can be used to suggest possible mappings.



The data aliases of the dictionary and data set are searched separately.

Selecting a data alias and clicking *Hint* will search for the name of a data alias in the opposite list that matches the highlighted name.

Data Format: Sample1

Publication Data Set: "1000" (Account Info)

Data Fields

All or part of the Data Field name: Cable

☑ Include Record ☑ Include Mappe
☑ Include Fields ☑ Include Unmap
☐ All Field Types
☑ String ☐ Number ☐ Currer
☐ Date ☐ Time ☐ Intege
☐ Match case

5 Data Fields found

Name	Type	Parents
Cable Service	Record	"1000" (A
Cable Rate	String	Cable Ser
Cable Box No	String	Cable Ser
Cable Flag	String	"1000" (A
Cable LookUp	String	Cable Ser

Data Set

All or part of the Data Field name: Cable Box No

☑ Include Records
☑ Include Fields
☐ All Field Types
☑ String ☐ Number ☐ Currency
☐ Date ☐ Time ☐ Integer
☐ Match case

1 Data Fields found

Name	Type	Parents
Cable Box No	String	Cable Service

▶ **To change the data format for a data map:** open the data map you want to modify and the Data Map Editor is displayed. Click the *Data Format* browse button ... and browse to a different data format. Now follow “To create a data map” from step 4 onwards.

▶ **To change mappings:** highlight the field you want to re-map in the *Data Fields* list and it highlights the field it is mapped to in the *Data Set* list. The button becomes active, enabling you to un-map the fields. You can now re-map the data field.

▶ **To view data element properties:** right-click the element in the *Data Fields* or *Data Set* list and select *Properties*.

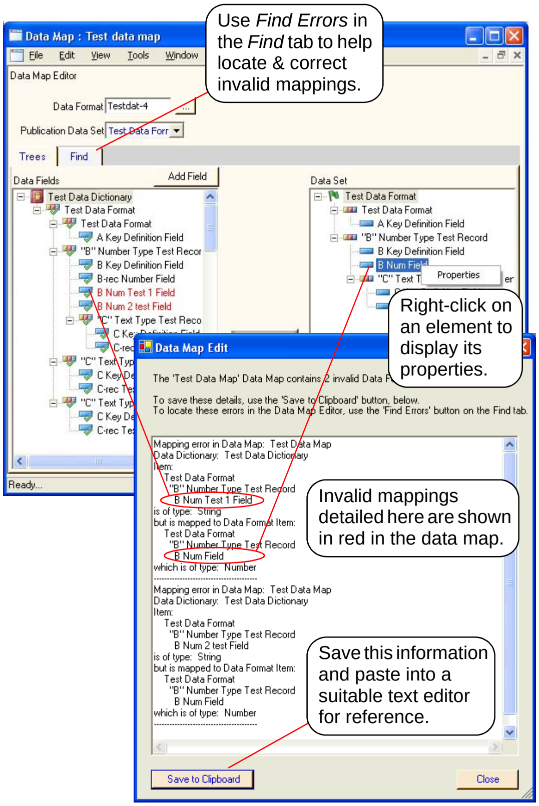
▶ **To search for data aliases:** select the *Find* tab in the Data Map Editor. If required, enter the name of the alias in the *Data Field name* field. Select the necessary parameters and click *Find*.

Invalid data map references

The data map can get out of step with the data format as the data evolves during the design – for instance if a field is renamed, its type changed, or if its position in the hierarchy is altered by moving it into a repeating group. This will cause invalid references in the data map.

These invalid references can be corrected automatically in the Work Center by using the *Update mappings for amended data format elements* option in the Data Dictionary and Data Map Generation Wizard.

You can also correct invalid references when editing a data map. Details of the invalid references are listed in a separate dialog box when the data map is opened. The data map editor will then be opened when this dialog box is closed, and you can correct the invalid references.



The 'Test Data Map' Data Map contains 2 invalid Data F

To save these details, use the 'Save to Clipboard' button, below.
To locate these errors in the Data Map Editor, use the 'Find Errors' button on the Find tab.

```
Mapping error in Data Map:  Test Data Map
Data Dictionary:  Test Data Dictionary
Item:
   Test Data Format
     "B" Number Type Test Record
       B Num Test 1 Field
is of type:  String
but is mapped to Data Format Item:
   Test Data Format
     "B" Number Type Test Record
       B Num Field
which is of type:  Number
-----------------------------------
Mapping error in Data Map:  Test Data Map
Data Dictionary:  Test Data Dictionary
Item:
   Test Data Format
     "B" Number Type Test Record
       B Num 2 test Field
is of type:  String
but is mapped to Data Format Item:
   Test Data Format
     "B" Number Type Test Record
       B Num Field
which is of type:  Number
-----------------------------------
```

Use *Find Errors* in the *Find* tab to help locate & correct invalid mappings.

Right-click on an element to display its properties.

Invalid mappings detailed here are shown in red in the data map.

Save this information and paste into a suitable text editor for reference.

Working with data dictionaries

The data record and data field references created within publication designs are always added to a data dictionary object. The purpose of these dictionaries is to allow you to group such data aliases into reusable categories.

The most obvious categorization is to list the data elements available to a particular DOC1 job – i.e. those defined in the data format that describes the contents of an input data file. To assist with this requirement you can export a data dictionary (along with a data map) from an existing data format.

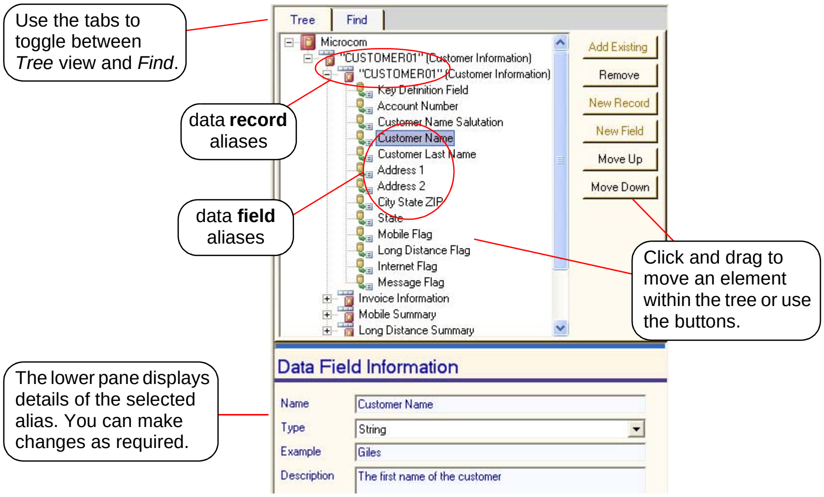
REFER TO "EXPORTING TO OTHER OBJECTS" ON PAGE 259 FOR FURTHER INFORMATION.

You may want to create other dictionaries to suit local requirements. Until you define custom data dictionaries a 'System dictionary will be assumed for all new data aliases that are created.

The data dictionary editor allows you to maintain existing dictionaries and to create new data aliases within a dictionary in advance of using them in publication designs.

In the editor the dictionary is presented as a tree structure and you can, for instance, place fields within records or nest records within other records. You may want to do so to reflect the structure of an actual input data file or to highlight other links between the data elements to publication designers.

IT IS IMPORTANT TO NOTE THAT THE STRUCTURE SHOWN WITHIN THE DATA DICTIONARY IN NO WAY INFLUENCES HOW DOC1 GENERATE SEES ITS INPUT DATA IN THE PRODUCTION ENVIRONMENT. THIS IS DEFINED SOLELY IN THE DATA FORMAT BEING USED WITH A PARTICULAR JOB.


Use the tabs to toggle between *Tree* view and *Find*.

data **record** aliases

data **field** aliases

Click and drag to move an element within the tree or use the buttons.

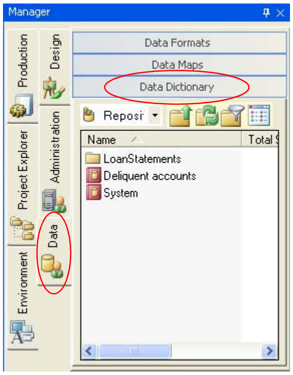
The lower pane displays details of the selected alias. You can make changes as required.

Data Field Information

Name: Customer Name
Type: String
Example: Giles
Description: The first name of the customer

▶ To create a new data dictionary:

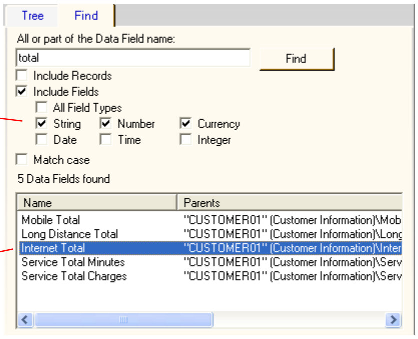
1. In the Work Center Manager select the *Data* tab and then click the *Data Dictionary* bar.
2. Select *File/New Data Dictionary* from the menu.



▶ To export a data dictionary from a data format: see “Generating a dictionary and data map” on page 262.

▶ To view or edit an existing data dictionary: double-click on the object in the Work Center Manager.

▶ To search a data dictionary: open a data dictionary and click the *Find* tab. Enter the name for the data alias, select the necessary search parameters and click *Find*.



All or part of the Data Field name: total [Find]

☐ Include Records
☑ Include Fields
☐ All Field Types
☑ String ☑ Number ☑ Currency
☐ Date ☐ Time ☐ Integer
☐ Match case

5 Data Fields found

Name	Parents
Mobile Total	"CUSTOMER01" (Customer Information)\Mob
Long Distance Total	"CUSTOMER01" (Customer Information)\Long
Internet Total	"CUSTOMER01" (Customer Information)\Inter
Service Total Minutes	"CUSTOMER01" (Customer Information)\Serv
Service Total Charges	"CUSTOMER01" (Customer Information)\Serv

Select the necessary search criteria to focus your search.

Selecting an alias from the search results pane highlights it in the *Tree* view.

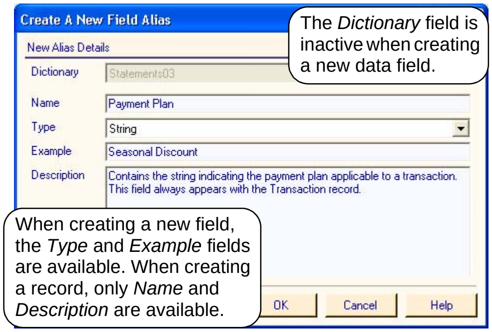
▶ To add a new data field or record to a dictionary:

1. Open a data dictionary.
2. If the new alias is to be nested – e.g. if you want to add a data field within a data record – click on the existing alias that will act as the parent.
3. Click the *New Record* or *New Field* button.
4. In the *Create a New Alias* dialog box give the new field a reference *Name* (which will be shown in selection lists) and add an additional *Description* if required.

5. For data fields:

    - Select a data *Type*; this is used to decide default output formatting for the field although it can be adjusted later. See “About data types” on page 104 for details.
    - Enter *Example* data; this is used as sample contents when the field is displayed in the editor.

6. Click *OK* to create the new alias.



Create A New Field Alias

New Alias Details

Dictionary: Statements03

Name: Payment Plan

Type: String

Example: Seasonal Discount

Description: Contains the string indicating the payment plan applicable to a transaction. This field always appears with the Transaction record.

The *Dictionary* field is inactive when creating a new data field.

When creating a new field, the *Type* and *Example* fields are available. When creating a record, only *Name* and *Description* are available.

▶ **To copy an alias from another data dictionary:** click on the appropriate alias if the new alias is to be nested and click the *Add Existing* button. Select the alias to be copied using the dialog.

▶ **To delete a data alias:** select the alias and click the *Remove* button.

▶ **To edit a data alias:** double-click on the alias and edit the settings as required.

Into Production

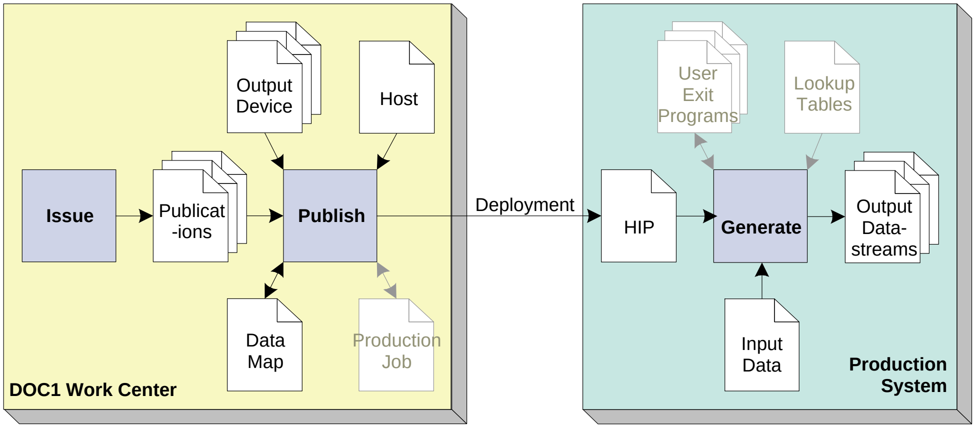
The production cycle

DOC1 Generate is the batch program that processes production jobs on your chosen host system. Generate reads information about the job requirements from a DOC1 HIP file, merges the input data file it is passed with your publication designs and produces output datastreams ready for printing or presenting on your intended output devices.

A production job is made up of one or more publication designs that share the same input data file. The appropriate publication is selected automatically when the control data with which it is associated (the *Start of publication* record as defined in a data format) is encountered in the input data.

Production jobs are prepared using a **Publish** task in the DOC1 Work Center. This brings together the selected publications and creates a single HIP file that will control the job in the batch environment. The HIP file contains all the information and resources required by a Generate to process the job and to print/present the output datastreams it creates on the actual output devices. The settings you specify when publishing can be saved in a **Production Job** object for reuse.

Before you can publish you will need to provide information about your intended printer/browsers in one or more **output device** objects. You will also need to create a **host** object that provides operating system and deployment information regarding the intended production platform.



If you have unresolved data aliases in your publication designs – i.e. references to data fields or records that have not yet been linked with actual data elements that will appear in the input data file to be used by Generate – you will be asked to map them as part of the publishing process. Such cross-references are stored in a **Data Map** object for future reference.

At the completion of a publishing task a HIP file is created and is deployed to the location indicated in the selected host object. If the necessary connectivity is available, deployment can automatically copy the HIP to your host platform using FTP or APPC connections (as defined in the Host object when the HIP was generated). Or you can deploy the file locally and manually copy to your production platform.

Once the production environment is prepared you can run the DOC1 **Generate** program on the host system. You will need to specify the HIP file that controls the publication as a parameter when starting the program.

Previewing a publication

Previewing allows you to run a publication through the DOC1 production engine within the Work Center itself without the need to go through the formal publication and deployment process. It produces a PDF version of the intended output and displays it in a Work Center window for your review.

The regular preview feature uses default production settings to preview output as quickly as possible. If required, an *advanced preview* feature can be used to customize the preview process so that you can include runtime options such as supporting e2 Vault, Message1, Content Author, lookup tables and customer support keywords. You can also edit the settings for the PDF output device. While there is only one regular preview, each publication can have a specific advanced preview associated with it.

> ⓘ Advanced preview uses a wizard which is very similar to the Publish wizard. Refer to the Publish Wizard in “Running a publishing task” on page 369 for more information.

The sample data assigned to the publication is used when previewing. This is in place of the input data that would normally be passed to DOC1 Generate in a true production run.

If you have unresolved data aliases in your publication design you will be asked to map them to actual data elements (as defined in a data format) just as you would with the regular publishing process. As with regular publishing you must have a data format for your publication in place before you can run preview.

▶ To run preview:

1. Select the Design tab and then the Publications bar.
2. Highlight the publication you want to preview.
3. From the *Tasks* menu select *Preview*.

4. Complete data mapping if required.

   If the publication has unresolved data aliases you will be asked to update or create a data map that supplies the appropriate linkages.

   See “About data mapping” on page 331 for details.

5. Review output.

   PDF output datastream is composed and presented in a separate window.

▶ **To run an advanced preview:**

1. Highlight the publication you want to preview and select *Tasks/Advanced Preview*.

   The Advanced Preview Wizard is displayed.

2. Create a production job for the advanced preview.

   Click *Select* and the *Select Production Job* dialog box is displayed. Select *Create New Production Job* and enter a name. Click *OK*.

3. Complete data mapping if required.

   If the publication has unresolved data aliases you will be asked to update or create a data map that supplies the appropriate linkages.

   See “About data mapping” on page 331 for details.

4. Configure the Advanced Preview Wizard settings as necessary.

   On the *Output Device* page, the only output device available is PDF.

   On the *Input/Output* page, you can choose whether to use the sample data assigned to the publication, or browse for a different one.

   For details of the other settings, refer to the Publish Wizard in “Running a publishing task” on page 336.

5. When you have completed configuring the Advanced Preview click *Finish*

6. Review output.

   PDF output datastream is composed and presented in a separate window.

▶ **To specify sample data:** open the publication and select *View/Publication*. Double-click the Publication object within the logic map to display the *Publication* dialog box. Use the *Sample Data* browse button to choose a suitable input file.

Output, media and resources

In the DOC1 Work Center an **output device** object identifies the type of data stream to be generated by a publication and allows you to specify any custom settings related to the output environment.

Pre-configured output device objects are provided that reflect the most common requirements for the environment indicated by your DOC1 license. You can override these to provide custom settings as required. One or more output devices will need to be indicated when you use the *Publish* task to put a publication into production.

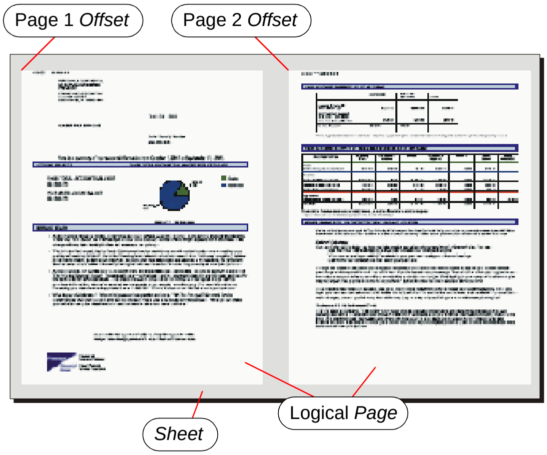
ⓘ DETAILED INFORMATION ABOUT SPECIFIC PRINTERS OR BROWSERS IS OUTSIDE THE SCOPE OF THIS DOCUMENT. REFER TO THE DOCUMENTATION PROVIDED FOR YOUR OUTPUT DEVICE IF YOU ARE UNFAMILIAR WITH ITS REQUIREMENTS.

Logical pages and media layout

The page layouts you design in the DOC1 Work Center are considered to be 'logical pages'. These are placed on the actual media pages – known as 'sheets' – using a defined offset for the top left corner. By default, a publication places a single logical page on each sheet and positions it as close as possible to the top left corner.

If you need to add multiple logical pages to a sheet you can use the *Layout* options of an output device object to customize the page layout. You can define how many logical pages are required per sheet and the sequence in which they should be used. Such media layouts are often known as '2-up', '4-up' or similar.

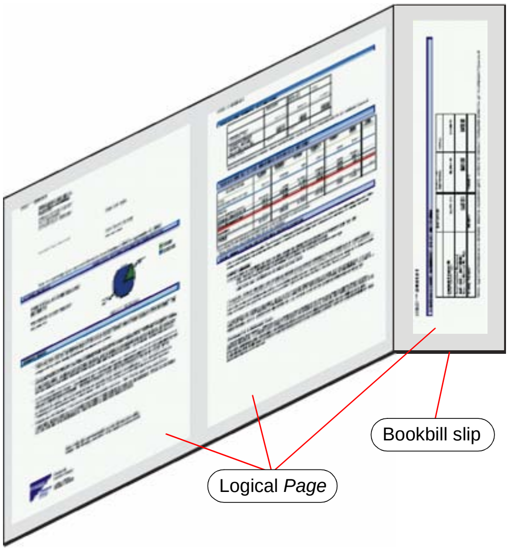
When working with multiple logical pages the intended sheet size becomes important. You will need to use the *Sheet size* output device options to let DOC1 know the area it has to work with when placing the logical pages.



'2-up' layout printed on landscape dimension sheet size

For physical output media DOC1 also assumes that only one side of the paper will be printed on. This is know as 'simplex' mode. If your printer supports double sided printing you can use the *Duplex* output device option to specify that both sides of the paper are to be used.

BookBill



Sheet 1 of an example *BookBill* layout

An additional option allows you to work with the specialized 'BookBill' media layout. This format assumes that a payment slip or similar is part of the first physical page to be produced by a publication design. The area reserved for the slip is in addition to the regular media layout – i.e. the first sheet produced has a larger area than any subsequent sheets.

When working with a BookBill format your publication design will need to output an additional logical page in the appropriate sequence that contains the required content for the payment slip. When you are creating output for a duplex process you will need to have two additional logical pages – one per side.

Font and image resources

By default, a publishing task converts the Windows font and image resources used when designing a publication into the format required by the target output datastream environment. These are stored in the HIP file along with all the other controls required by the publication in production environment.

Such resources will be created and referenced using file names that adhere to the requirements of the target output datastream. For PostScript, PDF and variants this normally means the resources can have the same names as used in the design environment. For datastreams that support a limited file naming convention (for instance, AFP and Metacode) the files will be created with alias names that primarily consist of a unique number reference. For example, an image resource of name 'Q3 results.bmp' may become 'S100014' when a HIP is created for AFP output. If required, you can use a **resource map** to override such defaults and indicate a specific name to be used for each individual resource file when it is being prepared for use with each type of output datastream.

You may also want to use a resource map to reference the names of existing resources in your output datastream environment. For example, if you already have a Metacode version of Helvetica 10pt on your printer (but have not imported it into the Work Center) you can indicate that all references to this font are replaced with the existing resource name.

REFER TO THE DOC1 PROGRAMMER'S GUIDE FOR DETAILED INFORMATION ABOUT CONFIGURING AND RUNNING DOC1RPU.

Unless you specify otherwise, font and image resources are always embedded into each document within the output datastream where they are used. Output device settings allow you to turn off automatic embedding and this can provide a significant reduction in the size the output file created by DOC1 Generate. However, you will need to consider alternative methods of making the resources available to the printer/browser environment if this is not already the case. The DOC1RPU utility is provided with DOC1 Generate distribution material to extract presentation resources from a HIP file if required.

Overlays

AFP and Metacode overlays can be included in your document design. As with other native resources, overlays must first be imported into the repository before they can be included in your document, and can only be inserted in the background layer. Overlays can be referenced in the print environment, or if you are working with AFP, they can be embedded in the publication HIP file. Note that embedding is controlled via the associated output device setting. See also “Overlays” on page 88.

Format of the output datastream

The default format of the logical records that make up an output datastream is set to cater for the most common requirements for each combination of production platform and output device type. However, you may need to adjust this format to cope with local circumstances, in particular if you need to transfer the datastream before it is actually made available to the output device. The *Format of logical records* setting in an output device object allows this customization.

Support for ideographic text

DOC1 supports text using the extended character sets required for ideographic languages such as Chinese, Japanese and Korean (sometimes known as ‘double-byte’ text) for several output datastreams. It is not supported for the following datastreams:

- Metacode
- PCL
- HTML
- RTF

Using these datastreams with publications that output ideographic text will produce unpredictable results.

Creating output device objects

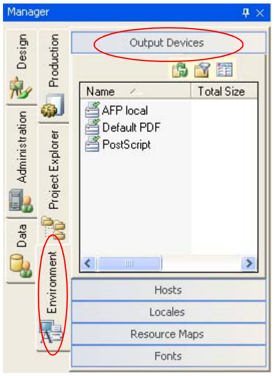
An output device holds the production settings for a particular type of output datastream as it relates to a target printer or browser. For example, you may have a group of settings for AFP 240 dpi printers, another for 300 dpi machines using the same datastream and settings for a local PostScript environment. You would need to create an output device object for each.

Every object has a range of general settings that apply to all output devices. These are supplemented by settings specific to the type of output datastream selected for the device.

▶ **To create an output device:**

1. In the Work Center Manager select the *Environment* tab and then click the *Output Device* bar.
2. From the File menu select *New Output Device* – an output device object is placed in the navigator.
3. Enter a name for the device and press Return. The *Output Device Configuration* window is displayed.
4. Use the Datastream type option to indicate the type of output that will be produced when this device is selected.
5. Configure other settings as required.

   Note that a wizards are provided for creating an IJPDS and MIBF output devices. After selecting the IJPDS or MIBF datastream type click *Configure* to invoke the wizard



▶ **To edit an existing output device:** double-click on the object in the Work Center Manager.

General output device options

Datastream type

Use this option to specify the type of output to be produced when this output device is selected.

Settings

The *Settings* tab contains options for the output datastream being produced. Options relevant for all datastreams are listed below. These will be supplemented with datastream specific options depending on the *Datastream type* you have selected.

Resolution

Each datastream has a default output resolution. This is used where the datastream requires a specific resolution be indicated within the protocol itself or where DOC1 needs to generate or reference bitmap images as part of the output. If the default is not suitable use this option to specify a different resolution to be used. This is specified as *Dpi* – dots per inch. The defaults are:
AFP – 240
Metacode – 300
PCL – 300
PDF and PDF variants – 72
PostScript and PostScript variants – 72
MIBF – 600

Image resolution

Use this option if the resolution of images generated for the publication needs to be different from other elements in the actual output datastream. Specify a *Dpi* – dots per inch value. The default setting of 0 (zero) indicates that the image resolution should use the regular *Resolution* setting.

Allow scaling up/down of images

These settings determine what happens if the resolution of images used in the publication design does not match the Resolution or Image resolution settings. If enabled, DOC1 will scale the images to the appropriate resolution when the publication is published (the scaled images are placed within the HIP file as always).

Embed fonts

By default, fonts are always embedded into each document within the output datastream where they are referenced. If the required fonts are already available to the printer/browser you may want to turn off this option to reduce the size of the datastream.

Embed images

As above but relates to graphics resources.

Default input tray

When working with printed media this setting specifies which input tray will be used to provide paper when the datastream is being printed until a different tray command is received. Note that the input tray and output bin being used can be dynamically modified during job processing. See “Controlling printer features” on page 221 for details.
Available trays are defined using the *Media* tab. The default is *Continuous Feed* which means either no input trays are available or that the default tray should be used.

(continued)	**General output device options**
Default output bin	As above but relates to the bin to receive the pages produced by the output datastream (where applicable).
Method for underlining	Where text is to be underlined this setting specifies the drawing method that will be used. *Native* – indicates that the underlining method within the output datastream protocol will be used if possible. This is the default. *Emulate* – when underlining is required an independent line is drawn under the appropriate text. Note that if the protocol cannot support underlining in a reasonable manner (as with LINEDATA output), then this setting is ignored.
Format of logical records	This setting allows you to customize the structure of the logical records that make up the datastream. You can either select from a range of keywords that format for a range of known scenarios or program your own logical record structure using formatting codes. These options are detailed in the DOC1 Programmer's Guide.

Media

The *Media* tab enables you to configure the printer modes, input trays and output bins for output devices that support these features.

Input Trays — Input trays are used by the printer to provide paper for the output datastream. Configure the necessary input trays by selecting them from the available input trays in the list.

Output Bins — The output bins provide the location where the pages are sent after being printed. Configure the required output bins by selecting them from the list.

Layout

The Layout tab enables you to configure settings necessary for placing multiple logical pages on a physical sheet of paper. The layout settings are dependent on knowing the size of the physical page, so if the output device's *Sheet Size* option is set to *AUTO* the layout options are disabled.

Sheet size — By default, the size of the physical sheets of paper used on the printer is assumed to be the same as the logical pages within your document designs. This is the *AUTO* option. Where this is not the case use this setting to specify the actual sheet size being used. You can choose from a range of standard sizes or select *CUSTOM* and enter specific dimensions.

Duplex processing — This setting determines if the datastream should be presented in simplex (*single-sided*) or duplex (*double-sided*) mode. Note that the duplex options can be modified dynamically during job processing. For printer drivers that support 'tumble' mode you can choose *double-sided tumble*. This causes the reverse side of the page – i.e. even pages (2,4,6 etc.) – to be output the opposite way round. See "Controlling printer features" on page 221 for more information.

(continued) **General output device options**

Pages Per Sheet

Select how many logical pages from your publication you want to output per side of sheet. The *Pages across* and *Pages down* options control the total number of pages for each sheet.

Book Bill remittance slip – this option enables you to include a remittance slip with each publication. The remittance slip is added by DOC1GEN to the end of the first sheet of the publication and does not affect the Sheet Size setting or any other output device option.

> For the *BookBill* option to function correctly you must include a remittance slip specific document in your publication. The page orientation for the document must be landscape and the page size cannot be larger than that specified by the Slip width and height settings. See “Using multiple documents” on page 99 for more details.

Placement Order

This setting controls the flow of the page placement on each sheet. The placement options are:
– *Pages flow left to right* or *Pages flow right then left*
– *Pages flow across and then down* or *Pages flow down and then across*
To ensure the publication prints correctly when using duplex printers the back side of each sheet is printed in the opposite order.
Place multiple publications on same sheet – this enables different publications to be placed on the same sheet of paper. A new publication will not be printed on the back side of a sheet of another publication. This option will not be available if *BookBill remittance slip* is selected.

Placement Details

These settings list the logical pages and on which *side* of the sheet they are placed. It also displays the *offset* for each page which is automatically set in the Layout tab.
It is assumed that each logical page is the same size. If you need to change the page size, double-click on the corresponding page and the *Page Details* dialog box enables you to change the settings.

> When a change is made that causes the Placement Details to be regenerated all the offset values are reset. For instance, if the sheet size, width, height or printer mode is changed in the Settings tab.

Example

This displays the layout of the publication’s logical pages on the physical sheets after Layout tab options have been set.

Resource mapping

Use this page to assign a resource map to the output device and to adjust the mappings it contains. For full details see “Using resource maps” on page 325.

AFP

The level of AFP features related to color and graphics will often vary depending on your specific printer/browser environment. The output device options for an AFP device contain a number of settings that allow you to control specific aspects of these features to suit your installation.

In an AFP environment, font, image and overlay resources are commonly stored in a structured file library for access by the software that handles the actual printing/presentation. If you choose not to embed the resources in the output datastream produced by DOC1 you will need to ensure that they are available in these libraries before the output can be sent to print.

File structure

By default, DOC1 generates AFP output with variable length logical records. Each logical record contains a single AFPDS structured field. Under z/OS these are also separate records within the file structure itself. On all other platforms the logical records form a continuous stream.

The maximum logical record length is normally dictated by the textual data stored in the PTX structured fields for a particular page. DOC1 Generate will attempt to place all text for a particular page within a single PTX structured field up to a default maximum of 8200 bytes. When publishing an application you can use an output device setting to specify a different limit if required.

Under z/OS

Ensure your output file has a record length suitable to receive the AFP datastream. If you are using the default *Limit record length* setting you should write to an output dataset with an LRECL of 8205**.**

You can optionally store DOC1 created AFP datastreams in **VSAM** datasets of type *relative record* (RRDS) or *keyed sequential* (KSDS). When VSAM is used the individual composed pages are stored as entries in the VSAM table. This format may be useful if you intend to archive or post-process the pages.

To use this option you will need to specify the 'RRDSAFP' or 'KSDSAFP' keywords in the *Format of logical records* setting or by coding custom formatting parameters for this setting that include the $BR or $BK output descriptors. Refer to the DOC1 Programmer's Guide for further information about custom output formats. You will also need to allocate an appropriate VSAM dataset in your DOC1 Generate JCL. Note that DOC1 always assumes VSAM datasets with fixed length records.

Under OS/400

Direct your output to a suitable physical file of type *DATA. The default storage type of STREAM means that structured fields are stored as a continuous stream as normally expected by the printing software.

Output device options for AFP

Settings

Printer carriage control

By default, the default printer carriage control (PCC) byte is not encoded. Use this option to select a coding system to be used for the PCC if required.

Color support

Use these options to modify the color commands used within the datastream.
Full – use the highest level of color references supported by the output datastream. Note that if your intended output device does not support color it may not be able to accept a datastream generated with this setting.
Downgrade – color references are restricted to the 16 color model used in AFPDS prior to the introduction of full color support. Color instructions within document designs will be mapped to the nearest equivalent color in this model.
24bit RGB – use this option if your output device supports the RGB color format. Color instructions within document designs will be mapped to the nearest equivalent RGB color.
32bit CMYK – use this option if your output device supports the CMYK color format. Color instructions within document designs will be mapped to the nearest equivalent CMYK color.
None – all color instructions within document designs are ignored. No color references are used in the output unless required by the datastream format in which case black is always used.
Note – if your publication uses rounded corners on a shaded table, choosing 24bit RGB or 32bit CMYK will ensure consistency of color shades in your publication.

Create fonts

Create AFP fonts from the Windows fonts used in your publication. These can then be either embedded in the output datastream – see “Embed fonts” on page 278 – or extracted manually to the output device.

Color image compression

If your publication uses many large images, you may wish to compress them to make the output datastream smaller. Use this option to specify the level of compression for color images. The default is best quality – *None* – i.e. the image is not compressed.

Embed Overlays

Selecting this option will embed the overlay in the publication HIP file. If you wish to reference overlays stored in the print environment then this option must not be selected. The default is not to embed overlays.

(continued)	**Output device options for AFP**
Graphics structured field size	If Level 2 GOCA is specified the amount of data stored in GAD structured fields may become very large. Use this option to specify the maximum size (in bytes) to be used for such fields before a new field is generated.
Limit record length	This setting indicates the largest permitted size of a PTX structured field within AFP output. PTXs contain the document text and typically produce the longest logical records in the output file. You may want to adjust the default if your production system is z/OS where the size of logical fields is relevant to dataset attributes.
Level 2 GOCA	When drawing charts or tables with rounded corners DOC1 uses GOCA instructions within GAD structured fields that conform to recent versions of the AFP architecture. This may cause printing difficulties in some AFP environments particularly where AFCCU controllers which have a version lower than 9.6 are in use. If required set this option to YES to cause DOC1GEN to use earlier versions of the GOCA instructions. Note that this may cause a significant increase in the output file size.
Use IOB for image rotation	By default, DOC1 uses the AFP Include Object (IOB) command, which allows images to be automatically rotated when landscape pages are being produced.
Color management	Selecting this will embed an image's ICC profile in the output datastream in CMR (Color Management Resource) format. This can improve the color quality of the image on printers that support it.
Max pels for thickest line	This setting indicates the maximum number of pels to be used when drawing lines and boxes. In AFPDS thick lines (rules) are actually drawn by using a number of lines at adjacent positions. The maximum thickness of any one line varies from printer to printer. Optimizing this setting for your printer and application can significantly reduce the size of AFP file generated by DOC1.

(continued) **Output device options for AFP**

Output font format

Determines the type of AFP font resource to be created. Options are:

AFP Bitmap (FOCA) – fonts are created as native AFP bitmap fonts (non-scalable) in accordance with the AFP FOCA definition. This option produces fonts suitable for most AFP print environment but note that file sizes will be very large where DBCS output is required.

TrueType (DOF) – TrueType or OpenType fonts used with the publication design are included directly in the output datastream within an object container. While this will produce much smaller file sizes than the bitmap option, note that your AFP printer must specifically support such Data Object Fonts (DOF).

AFP Outline (FOCA Type1) – fonts are created as native AFP FOCA Type1 fonts (Embedded Type1 Outline fonts) in accordance with the AFP FOCA definition. This option produces fonts suitable for most AFP print environments

Note that imported AFP bitmap fonts will not be converted to outline format.

Allow medium maps

The default option allows medium maps to be embedded into the AFP datastream. Clearing the check box creates output without a medium map, regardless of the settings in the *Medium Map* tab.

Use extended medium map

The default option includes input tray and output bin settings with medium maps when they are embedded into the AFP datastream. Clearing the check box results in the medium map's input tray and output bin settings being ignored.

Image Mode

This setting is used to ensure that images are printed correctly. The default option is *Black/White* (FS10). Use the *Color* (FS45) option only if your target printer has full color support (i.e. greater then 16 standard colors). For printers with the relevant object container support, the *Use originals (object container)* option enables you to use JPG, EPS, PDF Page Objects, TIF images.

Build fonts for IBM OnDemand

If checked, the output datastream can be viewed in the IBM OnDemand environment. You need to Run DOC1ACU to extract the necessary control information for integration with IBM OnDemand, see the DOC1 Programmer's Guide for further information.

(continued) **Output device options for AFP**

Media

General

This setting contains the name of the default *Formdef* file provided with DOC1 – *F1G1DFLT*. This AFPDS printer file contains information about the printer and a collection of medium maps. The medium maps contain information on input trays, output bins and printer modes.
The *F1G1DFLT* formdef file is automatically generated and placed inline to the print stream. If another name is specified in the *Formdef* field then the formdef is not generated.

Medium Maps

This enables you to edit the medium maps for the default *F1G1DFLT* formdef file, as well as create and delete medium maps for new formdef files. Use the *Edit Mapping* dialog box to provide a *Name* for the map and select which *Input Tray, Output Bin* and *Printer Mode* will be used with it.

ⓘ MEDIUM MAP NAMES MUST BE IN UPPERCASE AND HAVE A MAXIMUM LENGTH OF EIGHT CHARACTERS.

Other options

See “General output device options” on page 278.

Metacode

Rather than generating line data to be interpreted on the Xerox printer, DOC1 generates Metacode datastream directly to optimize printing times but this does mean that some non-standard printer settings may be required. Additionally, Metacode printers typically support both online and offline methods of receiving data for printing. These issues mean that you may need to customize the structure of the output file to suit your printer and data transfer environment.

The default logical record format for Metacode varies depending on the platform on which DOC1 Generate is run. For z/OS an online connection with the printer is assumed and the Metacode is generated as a standard output stream. On other platforms tape transfer is assumed and the default format includes an additional length field (Intel style RDW) at the start of each logical record that is commonly required with tape generation software.

If these settings are not suitable for your printer environment you can to specify a different organization of logical records using the *Format of logical records* option in output device settings when publishing.

The name of a Job Descriptor Entry (JDE) and an associated Job Descriptor Library (JDL) to be used when printing Metacode output are included in the datastream by all DOC1 applications. The default JDE/JDL names are 'ONLINE' and 'DPLJDL' respectively. If this is unsuitable you can define the JDE/JDL names to be used within as part of the output device settings used when put your application into production.

JDL adjustments

You need to be aware of some specific JDL requirements when printing Metacode generated by DOC1. In the following topics the settings are identified by the JSL statements (the source language that generates JDLs) that generate the appropriate setting.

PCC Definition

Metacode contains a printer carriage control (PCC) byte in each record that makes up the output datastream. These is typically in position 1 of the records although this is configurable if required using the *Format of logical records* option in output device settings when publishing.

The values in the PCC byte indicate certain standard functions to the printer of which DOC1 applications use only two – *skip to channel 1* (typically generating a new page) and *print (no space)*. The Xerox printer can interpret a range of coding systems used to specify the PCC values and their associated functions. The most commonly used coding systems for carriage controls are 'ANSI' and 'Machine Code'.

The JDL used when printing Metacode output produced by DOC1 applications will need to be aware which coding system has been used in the print datastream. By default, ANSI format PCCs are used in Metacode output generated by a DOC1 application. This is typical of the format used by Xerox off-line configurations. To use this format, your JSL should contain the following statement:

```
LINE PCCTYPE=ANSI,VFU=1;
```

You can select the coding system to be used for the PCC in output device settings.

Under z/OS you must ensure that the carriage control type indicated in the RECFM of the dataset receiving the Metacode matches the PCC type specified in output device settings.

Data Type

Xerox printers typically receive line data from applications and in these circumstances need to know the character coding system on which the data is based. However, Metacode from DOC1 applications is binary data that needs no interpretation by the printer. To relate this to the printer use the following JSL statement:

```
VOLUME CODE=NONE;
```

Output Statement

The JSL OUTPUT statement can be used to specify a variety of controls related to the appearance of the printed output and you should customize these as normal for the requirements of your application. However, all Metacode produced by DOC1 applications requires the following control to be set:

```
OUTPUT GRAPHICS=YES;
```

As VSAM

Under z/OS you can optionally store DOC1 created Metacode output in **VSAM** datasets of type *relative record* (RRDS) or *keyed sequential* (KSDS). When VSAM is used the individual composed pages are stored as entries in the VSAM table. This format may be useful if you intend to archive or post-process the pages.

To use this option you will need to specify the 'RRDSMTC' or 'KSDSMTC' keywords in the *Format of logical records* setting or by coding custom formatting parameters for this setting that include the $BR or $BK output descriptors. Refer to the DOC1 Programmer's Guide for further information about custom output formats. You will also need to allocate an appropriate VSAM dataset in your DOC1 Generate JCL. Note that DOC1 always assumes VSAM datasets with fixed length records.

Output device options for Metacode

Settings

Color support

Use these options to modify the color commands used within the datastream.
Full – use the highest level of color references supported by the output datastream. Note that if your intended output device does not support color it may not be able to accept a datastream generated with this setting.
None – all color instructions within document designs are ignored. No color references are used in the output unless required by the datastream format in which case black is always used.

Graphics Spot Color

This option enables you to designate the spot color name used in the metacode img header. If the color name does not match any of the existing values in the datastream then the name is copied into header and all in-line graphic images are generated as spot colors.

JDL name

Indicates the name of the Job Descriptor Library (JDL) to be referenced when printing the output.

JDE name

Indicates the name of a Job Descriptor Entry (JDE) associated with *JDLName* that will be referenced within the Metacode output.

Optimize for printing

If this setting is enabled the Metacode will be organized so that printing is as efficient as possible. Not enabling this option may produce a minor saving on the amount of time it takes for DOC1 to actually generate the datastream.

Online PCC

If this setting is enabled a 'new page' PCC byte (`x'F1'`) will be placed in Metacode datastream before the start of any output records. This has been found to be required on some Xerox printers for correct operation.

DJDE token

This setting indicates a DJDE token name that is to be inserted into the Metacode output to request printer specific actions. The token must be coded as a single character representing its format followed by the token string itself enclosed in quotes. The maximum length of the token is 29 characters/bytes.
The possible formats are:
A – the token name is included in the datastream as ASCII text, e.g. `A'token name'`
E – the token name is included in the datastream as EBCDIC text, e.g. `E'token name'`
X – the token name is coded as hex and will be included in the datastream as the appropriate binary values, e.g. `X'A1D390B2B1'`.

Job terminator

If required, this setting allows you to specify a custom value to be used to terminate the Metacode datastream. This must match the terminator specified in the JDL being used when printing the Metacode file. This value must be coded in hex notation, for instance: `x'A1D390B2B1'`. The default setting is the standard Xerox job terminator.

(continued)	**Output device options for Metacode**
Printer carriage control	By default, the default printer carriage control (PCC) byte is ANSI encoded. Use this option to select a different coding system to be used for the PCC if required.
Alignment offsets	This setting allows you to fine tune how output is positioned in relation to the actual physical pages produced by the output device. The offsets are specified as 'scan' and 'dot' values indicating adjustments to the top and left of the page respectively. These must always be specified as two 4-digit numbers separated by a space.
Media	
Input Tray	Initially the input trays are mapped to a default value. To override a value double-click on the input tray.
Output Bins	Initially the output bins are mapped to a default value. Double-click the output bin to change the setting.
Other options	See “General output device options” on page 278.

PCL

If you need to transfer PCL files between operating systems be aware that, by default, DOC1 applications generate such output as a true stream (as normally expected at the printer). For instance, if receiving a PCL file from z/OS onto a PC workstation you should not specify the CRLF attribute.

Positioning of the text can vary from printer to printer when non-zero orientations are used. You may need to adjust the position of text in relation to the bottom/right side of the logical page to cope with such variances.

Output device options for PCL

Settings

Pels offset from left of physical page

The number of pels by which printing is to be offset from the left side of the physical page.

Pels offset from top of physical page

The number of pels by which printing is to be offset from the top of the physical page.

Adjust right anchor position

Specify the number of pels by which orientated text is to be adjusted in relation to the right side of the logical page.

Adjust bottom anchor position

As for *Adjust right anchor position*.

Use non-HPGL/2 commands

The PCL commands used to draw charts are, by default, in HPGL/2 format as supported by PCL5 printers. Select this option if your printer does not support HP/GL2 and your publication does not use charts (other than 2D bar charts). Note that if your publication uses advanced chart features and this option is set, DOC1GEN will abort.

Output font format

The default *Bitmap* will include a separate font for each different point size used. *TrueType* will include a single font to represent multiple point sizes of the same font, which can reduce the size of the PCL output file.

Image mode

The default *Color* option should be used if you are using a color printer. If a monochrome printer is being used you must select the *Black/White* option.

Other options

See “General output device options” on page 278.

PostScript

PostScript is a text language and by default, DOC1 generates the PostScript commands in a format suitable for reading on the production system. If you are generating PostScript output under z/OS or OS/400 this means that the PostScript commands would be in EBCDIC-based text and any referenced resources in ASCII-based text. To avoid having multiple character codes in the same file, the default code page for PostScript is the ASCII-based *Windows 1252@ISO Latin1*.

Where this is not the case or in other non-standard scenarios you can override the default setting and indicate that a specific code page be used for the PostScript command text. Additionally, to cope with variations in the way systems interpret non-alpha numeric characters, you can indicate specific code points to be used for the PostScript control characters. Both of these options are defined in output device settings when publishing.

On some platforms you may also want to restrict the size of the PostScript records being generated.

Some external print management systems require a page number to be used within the PostScript page descriptor command. Output device settings allow you to choose between using a job or document page number for this feature.

DSC comments

DSC comments are typically used to communicate with PostScript document management software. DOC1 allows you to add such comments to the output datastream using an action object within your publication logic.

SEE "POSTSCRIPT DSC" ON PAGE 222 FOR DETAILS OF CREATING A DSC COMMENT ACTION.

Note that DSCs are not validated by DOC1 – you must ensure that the comment syntax you enter for the action conforms to the requirements of your document management environment.

Resource embedding

Normally DOC1 will place resources to be embedded into a PostScript file within the file header or 'Form Space'. The exception to this is EPS images which, by default, are embedded directly in each page where they are required; this is known as **inlining**. This is done to reduce the memory overhead on the printer which can be significant when EPS files are included in the PostScript file header. However, you can choose to embed some or all EPS files in the Form Space if required.

Do this by running the DOC1RPU utility against the HIP file that controls the application. Use the *MarkForFormSpace* option to specify the resources that are to be included in the header.

> Refer to the information about RPU in the DOC1 Programmer's Guide.

Image caching

Images intended for PostScript output can optionally be cached in printer memory either on an image by image basis or for all images, as required. This is done to increase print performance. By default, images are not cached.

▶ To cache images:

1. Select the required PostScript output device from the *Environment* bar in the Work Center Manager.
2. Click on the *Resource Map* tab and view the images defined for the current resource map.
3. Select the image you wish to cache and click on the *Edit* button.
4. Use the *Mappings Details* dialog box to enable the *Cache image* option.

> You can set all images to be cached or not cached by clicking the *Cache All/Cache None* buttons on the *Images* tab.

See "Using resource maps" on page 325 for further information on working with resource maps.

Printer resident images

Images that are used in your publication may also be resident on the target printer and can be used from there. The image would then be referenced by name, rather than the actual file being included in the resource pack. This can result in a smaller HIP file.

> You must ensure that the image is available on the printer in the same format and size. This is outside the scope of DOC1.

▶ To specify a printer resident image:

1. Select the required PostScript output device from the *Environment* bar in the Work Center Manager.
2. Click on the *Resource Map* tab, select the image that you want to use from the printer and click on the *Edit* button.
3. Use the *Mappings Details* dialog box to *Map to a printer resident image*.

Output device options for PostScript

Settings

Font subsetting

Only those characters that are used in a document are embedded in it. This reduces the size of the output file produced by DOC1 Generate, particularly when DBCS fonts are used

> Subset fonts are not suitable for use with PCE or other post-processing tools.

Output code page

If the default formatting of the PostScript commands is unsuitable for your environment use this option to indicate a specific code page to which PostScript will be translated before being output.

Record length

Specify a maximum record length for the PostScript commands in the output datastream. This should be an integer in the range 64-255.

Page label

Specifying JOB means that the current page number within the complete output stream will be used as the parameter in the PostScript page descriptor command (%%Page). This is the default setting.
Specifying DOCUMENT means that the current page number within the active document will be used as the parameter.

Customizations

The default is *None*, while selecting *Truepress Jet520* makes the Postscript output compatible with that specific printer.

Hex Encode output strings

In certain circumstances text in PostScript output is, by default, hex encoded, for example where non-Western languages are being output and for all text when DOC1 Generate is run on an EBCDIC-based operating system. You can use this option to override these defaults and force DOC1 always to output regular text. This may be required where a post-composition process needs to manipulate the output but note that it is the users responsibility to ensure that this option is suitable for use with the production environment.

Preserve image color space

This applies to TIFF and JPEG images only. The default is off – which will convert images to BMP with RGB color space. Selecting the option will convert images to EPS2 and retain the original color space, e.g CMYK, RGB.

Color support

Use these options to modify the color commands for text and shapes used within the datastream.
24bit RGB – use this option if your output device supports the RGB color format. Color instructions within document designs will be mapped to the nearest equivalent RGB color.
32bit CMYK – use this option if your output device supports the CMYK color format. Color instructions within document designs will be mapped to the nearest equivalent CMYK color.

(continued)	**Output device options for PostScript**
Output font format	This option is used to specify which font format you want use for the production job when it is sent to the postscript printer. *Automatic* – fonts are converted to Type42 fonts. *Type1* – fonts are converted to Type1 fonts
Media	
Input tray	Enter the necessary *Job DSC* and *Reference DSC* for the input tray by double-clicking the tray to invoke the *Edit Mapping* dialog box.
Output bins	Enter the necessary *Job DSC* and *Reference DSC* for the output bin by double-clicking the bin to invoke the *Edit Mapping* dialog box.
Other options	See “General output device options” on page 278.

PDF

DOC1 Generate produces PDF that can be viewed in a PDF browser either as one file containing all documents generated by the application, or as a series of PDF files each containing a single document. You can also produce PDF as an output datastream which can be passed directly to the Document Repository in a DOC1 Archive environment, or for processing by the DOC1DIME utility for e-mail distribution. The *Document processing* output device option determines the type of PDF output produced.

PDF documents have a number of identification attributes that can be applied such as *password, email address, title, author, subject* and so on. They can also have *permission* attributes that determine what actions the reader is allowed to carry out with regard to a particular document such as the ability to print or edit. In the DOC1 environment these are defined as attributes of a publication object. Any attribute that is specified is automatically associated with every document produced by the publication in the production environment. Note that in order for *permission* attributes to be enabled within PDF documents you must also specify an owner password in the appropriate output device settings (see below).

▶ **To specify publication attributes:** open a publication design and switch to the publication view. Double-click on the publication object to display and amend its attributes.

ⓘ FOR MORE INFORMATION ABOUT PUBLICATION ATTRIBUTES SEE "PUBLICATIONS AND DOCUMENTS" ON PAGE 96.

Document security

DOC1 generated PDF documents can optionally be password protected and encrypted to protect them from unauthorized access. Two types of password can be applied: *user* and *owner*.

User passwords protect documents from unauthorized opening in a PDF viewer. When using this feature the passwords to be applied are specified as part of publication attributes. You should resolve the *Password* attribute with a value that provides the required password setting. The password can be varied from document to document if the value used is based on input data or other objects providing dynamic values.

Owner passwords protect the PDF documents from unauthorized modification after they have been created. This option is assigned globally to the documents produced by a publication and is specified in output device settings when it is published. For fully secure documents you can produce a randomized password to prevent any amendments to the PDF documents after they have been created.

Encryption is automatically applied to all documents that have either type of password specified – see "Encryption key length" on page 297.

Additional options in output device settings can prevent various specific activities being carried out against the documents when they are opened in a viewer or editing tool.

Resource handling

By default, text and graphics within PDF output are compressed using the LZW algorithm (as used by Adobe Acrobat). Alternatively you can set the output device settings to use the Flate compression algorithm, which typically results in a better compression but requires more processing time. You can also turn off compression altogether, or choose to exclude embedded images from compression if required.

Image resources other than JPEG files are automatically converted to bitmap format and therefore can be compressed as above. JPEGs are passed to the output stream unchanged and therefore it is the user's responsibility to make sure that the appropriate compression is applied before the files are used in Designer.

ⓘ YOU SHOULD CONSIDER USING JPEG WHERE COLOR FIDELITY OR CONTROL OF FILE SIZES IS IMPORTANT TO YOUR APPLICATION.

As PDF output is likely to be distributed electronically it is a common requirement not to embed fonts so that the file size is kept as small as possible. PDF viewers (in common with PostScript printers) will normally have a basic set of fonts available – known as the Base 14 group – and will attempt to match missing fonts using the font references and attributes within the PDF. Use the *EmbedFonts* setting in the General output device section as appropriate – see page 278 for details.

Output device options for PDF

Settings

Font subsetting

Only those characters that are used in a document are embedded in it. This reduces the size of the output file produced by DOC1 Generate, particularly when DBCS fonts are used.

Document processing

Use this setting to determine the type of the PDF output produced by DOC1 Generate. Options are:
Compound – a single file containing all documents produced by the application. This can be passed to directly to a DOC1 Archive or be processed by the DOC1DIME utility to create individual PDF documents for e-mail distribution. Note that the file produced by this option cannot be viewed directly using a PDF reader.
Merged – a single viewable PDF file containing all documents produced by the application.
Split – viewable PDF files created for each document generated by the application. If you choose this option you will need to specify a filename template for the output produced. Refer "Specifying a file template" on page 324 for further information.

Translate to 7 bit

If this option is enabled any image data and compressed text in the output will be translated to standard ASCII values to ensure the maximum amount of readable data. Otherwise such objects are output as true 8-bit values.

(continued)	**Output device options for PDF**
Compression	This setting determines how the output is to be compressed. Options are: *Fastest* – use the LZW compression algorithm. In most situations this results in a faster compression than the *Smaller* option. This is the default setting. *Smallest* – use the Flate compression algorithm. In most situations this results in a smaller output file than the *Faster* option. *None* – do not compress the output.
Compress images	If this setting is disabled all embedded images are excluded from compression. Note that if 'Compression' is *None* then this option is ignored.
PDF/A compatibility	This ensures that the output is compliant with the PDF archiving standard PDF/A-1b. Images in the publication will be converted to the same color space as the *Color Support* option, when necessary. e.g. if *24bit RGB* is selected then a CMYK TIFF image is converted to an RGB JPEG image. Note that the color space of an external keyed image will not be converted, which may result in incompatible PDF/A output. The *Embed Fonts* option must be selected; the *Compression* option is automatically set to *None* and the *Document Security* option is automatically disabled. When the *Document processing* option is set to *Compound*, DOC1DIME must also be run to make the output PDF/A compatible.
Document security	Allows various security options to be set to protect documents from unauthorized access – see “Document security” on page 295 for more information.
Owner password	If the owner password feature is required use this setting to specify the password string or use the *Random* keyword to instruct DOC1 to produce the password using randomized characters. If you specify the password you must enter a 32 character alphanumeric string. If you do not provide an owner password, any permission attributes specified for a publication will be ignored.
Allow printing	If this setting is enabled the PDF documents may be printed from the viewer.
Allow modifications	If this setting is enabled the PDF documents may be modified by a suitable PDF tool after they have been generated.
Allow notes	If this setting is enabled notes may be added to the PDF documents using a suitable PDF tool after they have been created.
Allow copying	If this setting is enabled the contents of the PDF documents may be selected and copied to the clipboard in the viewer.
Encryption key length	Encryption is enabled on all documents that have either type of password specified. The key length can be 40 (default) or 128-bit. Note that 128-bit encryption is not supported by Acrobat Reader versions earlier than 5.0.

(continued)	**Output device options for PDF**
Color support	Use these options to modify the color commands for text and shapes used within the datastream. *24bit RGB* – use this option if your output device supports the RGB color format. Color instructions within document designs will be mapped to the nearest equivalent RGB color. *32bit CMYK* – use this option if your output device supports the CMYK color format. Color instructions within document designs will be mapped to the nearest equivalent CMYK color.
Other options	See “General output device options” on page 278.

E-mail distribution

The DOC1DIME program is used to e-mail documents from within a PDF output datastream using a local e-mail client. It can only be used with PDF output produced with the *Compound* document processing option.

ⓘ THE MAILING FUNCTION OF DIME IS ONLY AVAILABLE UNDER WINDOWS AND IS A SEPARATELY LICENSED DOC1 FEATURE. IT INTERFACES TO THE E-MAIL CLIENT USING EITHER CMC (COMMON MESSAGING CALLS) OR MAPI (MESSAGING APPLICATION PROGRAMMING INTERFACE) APIS.

The e-mail addresses to be used must be specified as part of publication attributes. You should resolve the *Email address* attribute with a value that provides the address to be used for each document. Normally this value will need to be based on input data to provide individual addresses for each document.

DOC1DIME uses an initialization file (INI) to specify the DOC1 output file to be processed, where the output documents are to be created and any other processing options including e-mail settings. The program is then run as a batch job on the production system specifying the INI file as a parameter. other govern. Full information about using DIME including all INI settings, e-mail API details and platform specific guidance is provided in the DOC1 Programmer’s Guide.

RTF

DOC1GEN can generate RTF output which can be viewed and edited using an RTF editor, such as Microsoft Word. The output will be split into separate files for each publication produced by a job. You will need to specify a filename template – used to provide unique names for each file. See “Specifying a file template” on page 324 for more information.

The layout of an RTF document may not look exactly as designed in DOC1. This is because:

- RTF may be interpreted slightly differently by different editors;
- The RTF format does not support all the features offered by DOC1. Where possible, an alternative method of presenting the feature has been used.

Other restrictions to be considered when generating RTF output:

- **Inline objects**, other than in-line images, are not supported and will cause Generate to stop processing.
- Any **graphic with content**, for example, text box, header, footer, address block, reserved area, barcode etc. will be converted to a table with a single cell without borders.
- DOC1 inserts **physical page breaks** into RTF output where a new page is required. As a result, if you subsequently edit the RTF, the content will not flow automatically between pages.
- **Fully justified text** will be changed to justified text.
- **Character alignment** on anything other than a decimal point is ignored.
- The RTF driver is not supported for applications using message applications.
- No **security** settings will be applied to the generated RTF documents.
- **Charts** are created as a non-editable image;
- Fonts using **extended character sets** (as typically used in China, Japan etc.) are not supported.
- **URLs** will be converted to plain text.
- **Anchored shapes** will be placed in the correct position, but the anchor will be changed to a fixed anchor.
- **Bullets, paragraph numbering** and **page numbering** will be converted to plain text.
- Output is always **ASCII encoded** regardless of the platform used for Generate.

Output device options for RTF	
Word 97 compatible	RTF that is compatible with Microsoft Word 2000 is generated by default. Setting this option will generate RTF that is compatible with Microsoft Word 97.
Other options	See “General output device options” on page 278.

VIPP, VPS & PPML

These output devices are full color production printers where the size of the output datastream is potentially much larger than for other devices. As a result both can use image caching mechanisms to keep the size of the datastream to a minimum. The printers themselves can cache resources. Use the *Job name* option to place and reference image resources used by a DOC1 job within the printer cache.

DOC1 Generate can also make use of a local cache for holding images used by a job. This ensures that only those resources actually required to print a particular datastream are included in the output file. PPML is an open format that adds object reusability to a range of standard output streams. DOC1 currently supports PPML for PostScript and PDF output.

When generating VIPP, resources can be extracted from the application’s HIP file and transferred to the output device environment using DOC1RPU. Note that the option to extract and deploy resources for VIPP is only available when running DOC1RPU in a Windows environment. Refer to the DOC1 Programmer’s Guide for further information.

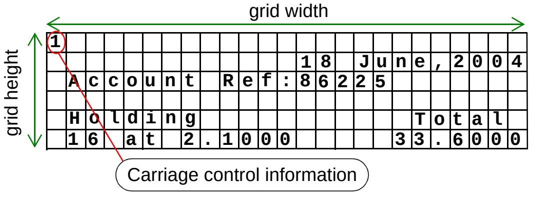
Output device options for VIPP, VPS & PPML	
Image cache mode	If the *Embed images* option is enabled this setting specifies how the images are to be stored in the output datastream. Options are: *All* – all images within the HIP file used with DOC1 Generate are added to the file header of the output datastream being created. This is the default setting. *Used* – a local disk cache is used to store those images that are actually referenced by a publication. The file to hold the cache is specified as the *Image cache file* when using the Publishing wizard to create a production job. At the end of job processing these resources are added to the file header of the output datastream being created. *None* – all images resources are placed inline in the output datastream within each document where they are needed.
Job name	If this parameter is specified and the *Embed images* option is enabled the image resources used by the job will be loaded into the disk cache of the actual output device. Subsequent jobs where *Embed images* option is not enabled will use the job name to access the relevant set of resources on the device.
Source type	PPML supports the generation of page objects in both PostScript and PDF format. Use this option to select the required type.
Position on whole	This setting applies to PPML output only. By default, page positioning offsets are specified as PEL values with up to three decimal places in the output datastream. Enable this option if the output is intended for a printer that supports only whole PEL values.
Other options	If the output device is a PostScript variant see “ Output device options for PostScript” on page 293. If the output device is a PDF variant see “ Output device options for PDF” on page 296. Also see “General output device options” on page 278.

Line Data

Line Data output is a textual representation of output from your application. It is intended for viewing via character based terminals or, as a feed for archiving. Note that graphical objects present in your design will be ignored in output generated by this driver.

Formatting

Output from your application is placed into a predefined grid providing an approximate representation of the original page design in terms of character positioning and justification.



```
1
             18 June,2004
 Account Ref:86225

 Holding           Total
 16 at 2.1000     33.6000
```

The dimensions of the presentation grid are set in the publishing phase as part of the output device settings for the Line Data driver. However, where lines exceed the grid width the height setting for the grid is automatically adjusted and extra lines are added to cater for the overrun.

Printer carriage control

The first column of the grid is set aside for printer carriage control (PCC) information. Note that the only control generated by DOC1 is the new page command on the first line of data for each page. The format of the PCC byte can be specified to conform to one of the following standard types: ANSI, ASCII, MACHINE or CRLF.

Output device options for Line data	
Height of character placing grid	This value indicates the height of the grid within which characters will be placed (i.e. the number of 'lines' available).
Width of character placing grid	The width of the grid within which characters will be placed (the number of columns available).
Output codepage	If the default formatting of the Line Data output is unsuitable for your environment use this option to indicate a specific code page to which Line Data will be translated before being output.
Printer carriage control	By default, the default printer carriage control (PCC) byte is ASCII encoded. Use this option to select a different coding system to be used for the PCC if required.
Force full width characters	If your application uses double byte character sets (DBCS) select this option to ensure that the spacing of characters within the grid is correct.

(continued)	**Output device options for Line data**
Configure for e-Messaging	Select this if the output datastream is to be used by e-Messaging. This will: – suppress new page commands from all but the first page of a document (these are normally generated for each page in a document); – include a publication ID after the new page command. Note that setting this option can make the Linedata output unsuitable for other uses.
Other options	See “General output device options” on page 278.

HTML

DOC1 supports two forms of HTML output.

The *HTML client/server* format typically supports applications that require interaction between web clients and a server on which pages and associated resources are stored. The output for each publication is created as one or more independent web pages. These pages and their associated image resources are output in an XML structure so they can be identified, extracted and deployed to the server environment as required. Cascading style sheets are used for presentation.

The *HTML e-mail* format provides a simplified version of HTML and is intended to be included as part of an e-mail communication with the end user. The output for each publication is created as a single HTML page (regardless of any page breaks in the design) and all images are simply referenced using the base name of the imported resource with a user defined URL. The HTML commands do not include cascading style sheets.

For both HTML formats you should bear in mind that only font references are included in the output and not the fonts themselves. We therefore strongly recommend that applications generating HTML use only fonts that are known to be generally available to web browser environments. Consider using a resource map if you Designer application does not use such standard font references (see “Using resource maps” on page 325).

Unlike other output devices DOC1 has no direct control over the font resources used for HTML output. Therefore when working with Arabic, Indian dialects or similar languages it is important to note that the browser or e-mail client is responsible for handling such things as inserting ligatures and formatting for right-to-left text.

HTML client/server

The following restrictions apply to DOC1 applications producing HTML client/server output:

- The use of the DOC1 barcode feature is not supported
- Rotated text – i.e. text elements presented other than at 0° – is not supported.
- The output is only compatible with browsers supporting cascading style sheets, such as the 4.0 or later versions of Internet Explorer and Netscape Communicator

The output device settings used when publishing are particularly important when HTML is being produced. They allow you to specify:

- Default browser presentation settings, including text colors and background.
- Location references for associated resource files that need to be deployed to a web server. Such references are embedded within the HTML documents and represent the locations where resources will need to be made available.
- Browser optimization options.

Note that the color settings may be overridden by the end user using browser customization options.

The HTML pages produced by a DOC1 job are created within a single XML file that also contains any other resources required to present the pages. It can be parsed to extract the pages as individual HTML documents or passed directly to the Document Repository if you are using e2 Vault or e2 Present to manage the output.

By default, the DTD describing the XML is stored as an integral part of the file but you can optionally request that it be extracted to a separate file if required. On record based systems (z/OS and OS/400) you should allow a logical record size of 4k to receive the XML file.

Images used in your design are converted to png format. By default, converted images are identified by their original base name together with a generated identifier. For example, if you have included an image called welcome.bmp it will be converted to welcome_g0001.png. If required, this name can be overridden using the resource mapping facility, refer to “Using resource maps” on page 325 for further information.

Output device settings for HTML

Java Applet Record length

By default, the record length for data to be sent to the DOC1 Graphics applet used for chart/shape drawing is set to 128 bytes. If required use this option to define a different record length.

DOC1 Interchange XML

If checked, the output datastream can be loaded directly into e2 Vault.

Image path

If coded, defines the location of graphics to be read in at the end of the job and embedded into the output datastream.

Presentation options

Background color

This is the color to be used for the background of all pages when displayed in a browser. The default color is white. If Background image is also specified the Background color will only be used while the image is loading.

Background image

This is a URL indicating an image to be used as the background for all pages. No image is used by default. For example:
`http://doc1/webres/logo.jpg`

Link color

If specified, the color indicated will be used to highlight all hyperlinks. IMPORTANT: this setting will override the colors used for such links within document designs.

Active link color

If specified, the color indicated will be used to highlight active hyperlinks. Note that this setting may be overridden in some browsers.

Visited link color

If specified, the color indicated will be used to highlight hyperlinks that have been visited. Note that this setting may be overridden in some browsers.

(continued)	**Output device settings for HTML**
Shade using white	By default, presentation objects that have a shading value of less than 100% use the HTML background color in conjunction with the object color to produce the shade effect (sometimes known as 'true color'). Enabling this option means that the background will be treated as 'paper' and white will be used in conjunction with the object color to produce the shade effect.
No applet message	If specified, this message is displayed if charts are used within a page but the DOC1 graphics applet is unavailable to the browser.
CSSP	Specifies the method to be used for positioning objects within the HTML. This can have a significant effect on download or presentation efficiency. Options are: *DIV* – use HTML DIV tag. The line breaks used with DIV tags will minimize the file size and thus the bandwidth required to download it. The client browser may take longer to render the page compared to where the SPAN option is used. This option is set as default. *SPAN* – use the HTML SPAN tag. This will optimize the commands so that they can be rendered by the client browser as quickly as possible. The file size will be bigger by two characters per positioning object required by the page as compared to using the DIV option.
Inline CSSP	By default, the CSSP structure is placed inline within the HTML. Disable this setting to have it embedded within a style sheet.
Style sheets	By default, the style sheet is embedded in the HTML page. EXTERNAL means that the HTML page references an external file for its style settings.
Comment out style sheet tags	Enable this option if you wish to comment out style sheets tags. This may be required if your application specifically needs to support older versions of some web browsers. Note that by default, style sheets are not commented out.
Z-Order	Can be used to control the Z order of objects. By default, text is placed on top of any other objects. If enabled, the Z ordering will reflect the order in which the objects were passed through the HTML driver.
Optimization options	
Optimize for rendering	By default, the HTML will be optimized for the most efficient download speed. Enabling this option will optimize the HTML such that it renders as quickly as possible on the browsers but the file size of pages will be increased.
Chart threshold	This option specifies the maximum inline size (in bytes) for parameters related to chart drawing. Where the parameters required for chart drawing fall under this limit they will be included as meta data within the actual HTML pages. Parameter lists exceeding this limit are created as independent objects within the XML file.

(continued)

Output device settings for HTML

Document Type Definition options

External DTD

By default, the DTD that describes the HTML PAK file remains part of the PAK file. In this scenario DOC1EDU is expected to perform the HTML extraction process. If enabled, the DTD is externalized to the location specified in the *DtdFile* keyword.

DtdFile

This setting must be specified if the *ExternalDTD* option is enabled. Specify a path and file name that will receive the DTD.

DtdReference

By default, the HTML PAK file will reference its externalized DTD at the location indicated by DtdFile. If you need to relocate the DTD during deployment – e.g where OS file naming conventions prevent a valid reference or where the web server is not directly available to the system running DOC1GEN. You can use this keyword to specify the location to be used to reference the DTD within the PAK file.

Server references

Resource URL

By default, the vector graphic used for lines and boxes are assumed to be available at the same location as the HTML pages themselves once the resources have been extracted from the XML file. If a different location is used this option allows you to specify the URL where the resources are located and will be used to reference them within the pages. The following codes can be used to include attributes from the XML structure as part of the path if required (you can use the same information when extracting files from the XML):

&pakUID – unique ID for the current job
&groupUID – unique reference to account group
&groupID – contains the *keyword* parameter from the attributes of the first document in the publication (if coded).
&docUID – unique reference to the current document
&docID – a reference indicating the sequence of the document within the group
&pageID – a reference indicating the sequence of the page within the document
&jdUID – ID specific to a chart data object if present.

Examples:

```
http://doc1/html/resource
http://doc1/html/resource/&pakUID
http://doc1/&pakUID/&docUID/&pageUID
```

(continued)	**Output device settings for HTML**
Image URL	By default, vendor supplied graphics are assumed to be available at the location indicated by *Resource location* or, if this is not coded, at the same location as the HTML pages themselves. If required, use this setting to specify a different URL where the image files are to be referenced within the HTML pages. The URL syntax is the same as for *ResourceURL*.
Applet URL	By default, the DOC1 graphics applet required for chart/shape drawing is assumed to be available at the location indicated by *Resource location* or, if this is not coded, at the same location as the HTML pages themselves. If required, use this setting to specify a different URL where the applet is to be referenced within the HTML pages. The URL syntax is the same as for *ResourceURL*.
Locate chart data by	By default, the Java data location setting is expected to indicate a URL for locating chart/shape data files. Enable this setting if you want to locate the files using a server side script (CGI, ASP, PERL, etc.).
Chart data URL	By default, files containing chart drawing parameters are assumed to be available at the location indicated by Resource location or, if this is not coded, at the same location as the HTML pages themselves. If required, use this setting to specify a different URL where the chart data files are to be referenced within the HTML pages. The URL syntax is the same as for Resource URL. If Locate chart data by… is enabled, use this setting to specify a server side query script. For example: `http://scripts/getchart.cgi?&jdUID`
Other options	See “General output device options” on page 278.

Working with the XML file

The output from a DOC1 job producing HTML output is an XML file that contains both the pages and any associated resources. If the output is to be managed by e2 Vault or e2 Present you can pass the XML directly to the Document Repository without modification. Otherwise you will need to use a method of extracting the material from the XML into individual files before it can be browsed.

If you intend to use the HTML in a web server environment you will need to parse the XML file, identify the relevant resources and extract them to the appropriate server locations. Files other than the actual pages will need to be made available at the resource locations specified within the HTML itself (and which can be customized using output device settings). Detailed information about the XML file format and guidelines for creating an extract routine can be found in the DOC1 Programmer’s Guide.

If you simply want to extract the files from the XML for local browsing or onward processing you can use the EDU (Extract and Deploy Utility) program which is provided as part of DOC1 distribution material in the production environment.

The DOC1EDU program uses an initialization file (INI) to identify the XML input file and to specify the locations to be used for the output files. An index file can be created which provides information about the document and page sequence which produced the HTML and cross-references for resource files.

Full information about using EDU including all INI settings and platform specific guidance is provided in the DOC1 Programmer's Guide. However, the following is a simple example of how you might configure and run EDU to extract HTML pages and resources on a Windows host.

Simple EDU example

Create an INI file:

```
<Files>
;HTML output from Generate (XML)
Input=doc1html.xml
;Location to receive HTML pages
PageFileMask=\html\%1.htm
;Location to receive resource files
ResourceFileMask=\htmlres\%1
;File to receive index file
IndexFile=\doc1\htmlindx.txt
```

- `<Files>` is a required section name.
- `%1` is a place holder for the file names that will be automatically created by EDU. These will sequentially number the page and resource files in the order they are created.
- Comment lines start with a semi-colon.
- Save the new file as `extract.ini` (in this example).

Run the DOC1EDU utility:

```
doc1edu ini=extract.ini
```

HTML for e-mail

The HTML for e-mail (eHTML) driver provides a simplified version of HTML which is primarily intended to be incorporated into e-mail. It aims to be compatible with clients that may have limited HTML handling capacity and as a result is restricted to HTML v3.2 commands by default. More sophisticated results can be produced by optionally lifting this restriction but in no circumstances does the device use cascading style sheets or Java Script.

The output for each publication is created as a single HTML page (regardless of any page breaks in the design). These are output as individual files according to a path template. See "Specifying a file template" on page 324 for more information.

ⓘ WHEN DESIGNING AN APPLICATION SUITABLE FOR HTML FOR E-MAIL IT IS RECOMMENDED THAT THE EDITOR PAGE IS DEFINED TO HAVE A HEIGHT LARGE ENOUGH TO ACCOMMODATE ALL EXPECTED OUTPUT ON A SINGLE PAGE.

Images other than those created for barcodes are assumed to be server resident and are simply referenced within the HTML using the base name of the imported resource file at a user defined URL. At publishing time image resources are not created for this device.

Barcodes are supported for this device and rendered as GIF images output as independent files according to a further template path name. Unlike other images the barcode GIFs are assumed to require physical packaging with the HTML itself as part of the e-mail system.

Once a production job has completed it is the user's responsibility to check the validity of any links in the HTML pages, package barcode GIFs with the pages if required and then actually include the pages in the appropriate e-mail messages. DOC1 itself does not support direct e-mailing for this type of output.

Please note the following when working with this format:

- The <Title> element for HTML pages is provided by the *Title* option found in the Attributes tab of the *Publication* dialog box. See "Publication attributes" on page 97 for more details.
- When the document design includes **tables** a separate table is created in the HTML output for each table row in your document. Line thickness of all such tables is set according to the properties of the first cell in the design table.
- When limited to HTML v3.2 **fonts** are referenced by size only. Without this restriction *font-family and font-face* attributes will be included. Font sizes are mapped according to the following table:

Designer font size	HTML font size
<8 pt	1 xx-small
8<10 pt	2 x-small
10<12 pt	3 small
12<14 pt	4 medium
14<18 pt	5 large
18<24 pt	6 x-large
24 pt +	7 xx-large

- Only inline positioning is supported. Objects designed for anchored or fixed positions will automatically be changed to inline when using this device.

Use of the following DOC1 Designer features are not supported for this device. Results may be unpredictable if they form part of a design used with the device:

- **Charts**
- **Lines** and **shapes**

- **Bookmarks**
- **Shading –** in the *Shape* properties dialog box the *Fill Color* option can be used, but you should not use the *Fill Shading* option.
- **Table attributes**, including the following:
  - **Spacing before** or **after** a table
  - **Indenting** of tables
  - **Internal cell margins**
- **Page Setup view logic** – there is no execution of logic contained in headers, footers or the background object, this includes features like reserved areas.
- **Page numbering**, **breaks** and **new page** actions
- Paragraph formatting, including the following:
  - **Justified text**
  - **Bullets** and **numbered lists**
  - **Tab stops**
  - **Line spacing**
  - **Widows** and **orphans**

> ⓘ DUE TO THE LIMITATIONS IN PARAGRAPH FORMATTING LISTED ABOVE, IT IS RECOMMENDED TO LEAVE ALL PARAGRAPH OPTIONS AT THE DEFAULTS PROVIDED WHEN CREATING A NEW PUBLICATION.

Output device options for e-mail HTML

Code page

This is the code page used to format the text of HTML commands themselves. The default, UTS-WIN1252, is the ASCII format for standard Western characters and is suitable for most Windows based browsers.

Character set

Use this option to specify the character set to be referenced in the HTML header record, this identifies the language encoding being used by text elements.

Background color

This is the color to be used for the background of all pages when displayed in a browser. The default color is white. If Background image is also specified the Background color will only be used while the image is loading.

Link color

If specified, the color indicated will be used to highlight all hyperlinks. IMPORTANT: this setting will override the colors used for such links within document designs.

Active link color

If specified, the color indicated will be used to highlight active hyperlinks. Note that this setting may be overridden in some browsers.

Visited link color

If specified, the color indicated will be used to highlight hyperlinks that have been visited. Note that this setting may be overridden in some browsers.

(continued)	**Output device options for e-mail HTML**
Image URL	Use this setting to specify a URL where image resource files are to be referenced within the HTML pages.

Barcode Image URL	This option is only applicable when the Generate Barcode for URL option is selected. By default, barcode images will be placed in the same location as the HTML output files and are referenced within the HTML pages using the MIME standard. When selected barcode images will be placed in the location specified for embedding in the HTML pages. For example:
	`http://doc1/html/resources/`
	`//g1abc/output/bc/images/`

Generate Barcode for URL	Select this option to output barcode images from your application at the destination specified in the *Barcode Image URL* setting. If this option is not selected, references to barcode images are placed in the HTML page included in an e-mail with the 'cid:' command, using the Multipurpose Internet Mailing Extensions (MIME) standard.
Generate HTML v3.2	Selecting this option generates output that strictly adheres to the standards for HTML version 3.2 as specified by the World Wide Web Consortium (W3C). Use this option to ensure that your output will be displayed correctly in e-mail clients, such as Lotus Notes and Microsoft Outlook. If you do not select this option the HTML output may include the font-family/font-face attribute, which is not supported by version 3.2. Note that it is not recommended that this option is used when a publication contains Arabic text, as HTML v3.2 does not fully support right-to-left text. For more information on HTML version 3.2, see http://www.w3.org/TR/REC-html32.

Barcode file template	This allows you to define a base file name to be used for barcode GIF images. It must be specified if your publication uses barcodes and must be in a format suitable for the host platform. Two parameters are available to help make the filename unique:
	For example:
	`Publication output filename = Out%1.htm`
	`Barcode file template = Bar%1-%2.gif`
	will produce the following GIF image names:
	`BarOut0000001.htm-001.gif`
	`BarOut0000001.htm-002.gif`
	
	`BarOut0000002.htm-001.gif`
	`BarOut0000002.htm-002.gif`

Use table to position output	This forces each page to be output in an HTML table element, which preserves the designed width of the page. The table itself will not be visible.

(continued)	**Output device options for e-mail HTML**
Alignment of table	If a table is used to position the output, this allows you to specify its alignment. The effect of this setting will only be apparent if the window used to view the output is resized beyond the width of the table. This option would normally be used where the direction of text is right to left, to preserve an overall look. Default is *left*.
Compound file for e-Messaging	Select this to generate a single file containing ALL documents. This makes it more efficient for processing by e-Messaging. Note that a compound file is not supported by HTML browsers.
Other options	See "General output device options" on page 278 and " Output device settings for HTML" on page 304.

IJPDS

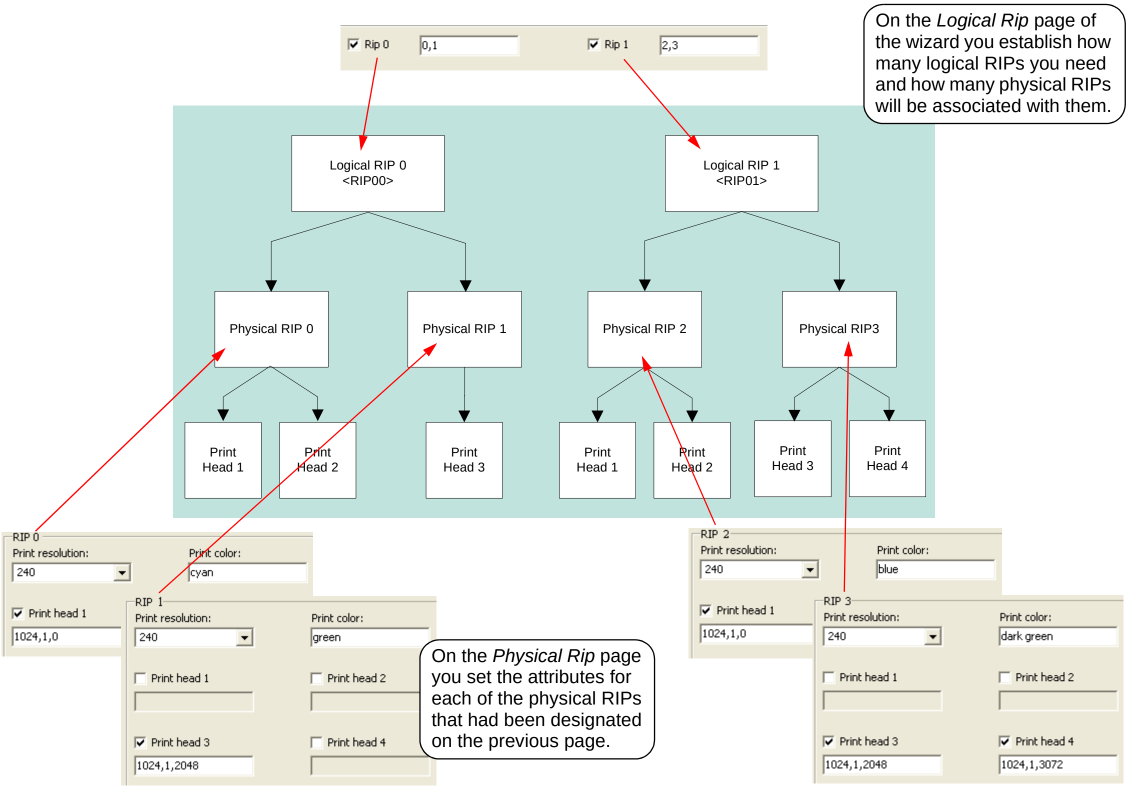
IJPDS output is designed to be printed on the Scitex family of high speed ink jet printers.

To enable the fastest possible throughput these printers rely on input from the datastream being processed to configure the print hardware environment. The IJPDS configuration wizard simplifies the process of setting the many options required to achieve optimum performance from your IJPDS printer. The wizard is automatically invoked when creating a new, or editing an existing IJPDS output device in the Work Center.

RIP configuration

IJPDS output generated by DOC1GEN is converted into bitmap page images and directed to the appropriate print heads by physical Raster Image Processors (RIPs) which also control printing over a designated width of the printer's page buffer.

Controlling Scitex RIPs and print heads



☑ Rip 0 0,1
☑ Rip 1 2,3

RIP 0
Print resolution: 240
Print color: cyan
☑ Print head 1 1024,1,0

RIP 1
Print resolution: 240
Print color: green
☐ Print head 1
☐ Print head 2
☑ Print head 3 1024,1,2048
☐ Print head 4

RIP 2
Print resolution: 240
Print color: blue
☑ Print head 1 1024,1,0

RIP 3
Print resolution: 240
Print color: dark green
☐ Print head 1
☐ Print head 2
☑ Print head 3 1024,1,2048
☑ Print head 4 1024,1,3072

On the *Logical Rip* page of the wizard you establish how many logical RIPs you need and how many physical RIPs will be associated with them.

On the *Physical Rip* page you set the attributes for each of the physical RIPs that had been designated on the previous page.

To allow for maximum flexibility, the concept of the logical RIP is used in the Scitex print environment. One or two physical RIPs sharing common attributes can be joined (stitched) and controlled by a single logical RIP. You specify the attributes of the print heads to be controlled by a logical RIP using these settings.

Color management

The color modes supported for IJPDS are black and white, spot and full color. When working in spot color you should try to ensure that the spot colors used in your document design are available on the printer hardware, if this is not the case then the closest color match is used during the print process. Refer to “Spot color printers” on page 220 for more details.

To maintain consistently high quality color printing you will need to periodically tune the color settings passed to the printer by DOC1GEN. The main reason for doing so is that print quality may vary due to changes in paper, ink and environmental conditions. To ensure consistent print colors you will need to associate color linearization tables to your output device based on the results of test print runs. These should be run on a regular basis at your print facility.

Dithering

IJPDS output requires all print objects to be connected to bitmaps. The colors presented in these bitmaps is generated by the dithering algorithm used by DOC1GEN – i.e. the routine that decides how the ink spots are to be mixed to form the required color or shade. These algorithms minimize the amount of ink used to print the application. So that performance versus presentation can be optimized for dithering operations you can individually select the algorithm to be used for text and external images.

You can, if required work with pre-dithered image resources, these must be converted using the supplied image conversion utility before use in the Designer, see the “Processing IJPDS and MIBF images” section in the DOC1 Programmer’s Guide for further details.

Configuration wizard settings

The IJPDS Configuration wizard allows you to define settings to be assigned to your intended IJPDS output device.

IJPDS wizard options

General settings

Print mode — This setting determines if the datastream should be presented in *simplex* (single-sided) or *duplex* (double-sided) mode. You also select how many logical pages you want to output per side of sheet, either *1–up* or *2–up*.

Color support — This defines if the printer is to expect *Full* (CMYK), *Spot* (black, plus spot color) or *None* (black only).

(continued)	**IJPDS wizard options**
Separate back-side color support	This setting allows you to define separate color support when printing in duplex mode. If this option has been selected you can then define if the printer is to expect *Full* (CMYK), *Spot* (black, plus spot color) or *None* (black only).
Text/Graphics/Image dither algorithm	Each of these settings allow you to specify the color matching process to be used with the object type. Options are: *RAST* uses basic raster dithering for text fonts. *FSB* uses the Floyd Steinberg error diffusion method for dithering text fonts.
Tumble mode	This allows logical pages to be output as ‘tumbled’ i.e. top to bottom. You can tumble *all pages*, *odd pages* only (1,3,5 etc.) or *even pages* only (2,4,6 etc.) The default is *No tumble*.
Max page size	This is the maximum number of printer dots available in the Y direction of the page buffer for the target printer. The default is 60 inches at the specified resolution.
Limit memory usage	This is the maximum amount of memory specified in MB that can be allocated by the driver before disk caching is used.
Print resolution	Indicates the dots per inch resolution of the print heads. On the target output device the default is *300 x 300*.
Page X & Y offset	The values specify the page origin *X* or *Y* offset in dots. This is needed for printing outside the cue and registration mark area and is based on the top left corner of the sheet being 0,0. Printing is shifted to the right for *X* offset and down the sheet for *Y* offset.
Cue marks	These settings allow you to define the size and position of cue marks to be printed on pages produced by the application. Cue marks are used to synchronize RIPs when printing in duplex mode. These values are expressed as a number of printer dots.
Allow for character compression	This setting determines whether character compression is used or not. The default setting is off, which is shown by the value of 1. Checking this option enables character compression and sets the value to 2.

Registration marks

These settings enable each page of the publication output to contain a print mark by the print heads individually when printing in full color mode, allowing alignment problems to be resolved. The values are specified as a number of printer dots.

Reg mark position	These settings must be specified as a pair to indicate the offset to be used for the top left of the mark. You can set between 1 and 4 registration marks.

(continued) **IJPDS wizard options**

Reg mark size

This setting indicates the size of all registration marks, which is a circle set within a cross.

Color attributes

These settings allow you to map the fixed color definitions available to your publication design with specific CMYK values to be passed to the printer. By tuning these setting you can more closely match DOC1 standard colors to those on your printer hardware.

Logical RIP assignments

This page contains a list of logical RIPs (Raster Image Processors). Logical RIPs contain settings about how print heads are to be configured.

Rip 0-15

These settings identify the logical RIPs to be used by the application. A maximum of 16 logical RIPS are permitted. Each logical RIP used must be assigned to one or more of the physical RIPs available on the target printer.

▶ **To make RIP assignments:** click on the *Set* button associated with the logical RIP, the *Physical RIP selection* dialog box is presented from which the required selection can be made.

Configure logical RIP attributes

These settings allow you to configure the attributes of each logical RIP defined on the *Logical RIP assignments* page. Click on the logical RIP from the *Logical RIPs* selection list to define the required attributes. Note that you also have the option to associate color linearization tables on this page when printing in spot color mode.

Print color

This option indicates the ink color the RIP is to work with based on either IJPDS, RGB, CMYK or DOC1 standard colors. It is recommended that this value is specified on the *Color attributes* page. Valid formats are:
RGB – R:n,n,n
CMYK – C:n,n,n,n
where n is a positive integer.
IJPDS Named Color – indicates a basic IJPDS color for full color operation, valid settings are:
Cyan, Magenta, Yellow and Black.
DOC1 standard Color – indicates DOC1 named colors, valid settings are:
N:Black, N:Blue, N:Brown, N:Green, N:Pink,
N:Red, N:Cyan, N:Yellow, N:Darkblue,
N:Darkgreen, N:Teal, N:Gray, N:Mustard,
N:Orange, N:Purple, N:White

(continued) **IJPDS wizard options**

Print head

These options allow you to enter values indicating the ink attributes of the print head on the target printer.

▶ **To define print head attributes:** Select the check box corresponding to the required print head and click the ... button, the *Print head attributes* dialog is presented, you can define the following attributes:
– the number of ink jets for the specified print head.
– the number of ink drops that form a printed dot.
– the relative position to the first print head.

Print position

This option enables you to specify that the RIP is to be used specifically for one or more of the logical pages specified by the *Print Mode* option on 'Page 1'. The available values are:
Front1 – front side, first logical page.
Front2 – front side, second logical page.
Back1 – back side, first logical page.
Back2 – back side, second logical page.

Set cue mark

Select this option to request a cue mark for the RIP. This is available for all color modes cue marks. Cue marks are generally printed using black print heads on the front side of the printed page. See “Cue Marks” on page 315

Print offset

The offset position of the RIP measured in pixels.

Font memory

This option is used to specify the available font memory of the physical RIP in MB.

Text/Graphics/Image color

These settings are only available when printing in spot color mode and allow you to associate color linearization tables to be used when printing text, line/boxes and images. Enter a path/file name or label conforming to the convention required for the host operating system. See “Color management” on page 314 for further information.

Color lookup tables

These settings are only available when printing in full color and black only modes, they allow you to associate color linearization tables for use when printing text, line/boxes and images. See “Color management” on page 314 for further information. Note that the path/file name or label entered on this page must conform to the convention required for the host operating system.
When printing in duplex mode you can set up color lookup tables for the front and back side of the page.

Black

Associates the linearization tables to be used when printing in black only and full color modes.

Cyan/Magenta/Yellow

Associates the linearization tables to be used when printing in full color mode.

(continued)	**IJPDS wizard options**
Black generation	Associates the linearization tables to be used with the black generation RGB to CMYK color conversion functions. Used in full color mode only.
Under color removal	Associates the linearization tables to be used with the under color removal RGB to CMYK color conversion functions. Used in full color mode only

File paths

This page is used to define path/filenames for IJPDS resource files used during production. The IJPDS output filename is created automatically using the production job name specified in the Publish Wizard or Production view.

Index file	Specifies the path/file name of the index file associated with the IJPDS output data file.
Object cache file	Specifies the path/file name of a temporary file to act as workspace for object caching.
Image cache file	Specifies the path/file name template of temporary files to act as workspace for caching images used by the application. Up to 16 files may be created for this purpose – %1 in the template is replaced with the file names to be used.

MIBF

The Memory Image Bitmap File (MIBF) format is a proprietary Miyakoshi Corporation file format for ink jet print streams, supporting high-speed, full-color production. To enable the fastest possible throughput these printers rely on input from the datastream being processed to configure the print hardware environment. MIBF produces multiple output files based upon the number of Bitmap Image Processors (BIPs) specified in your printer configuration.

In common with IJPDS output, color management techniques are based on the use of linearization tables and dithering algorithm, refer to “Color management” on page 314 for further information. Note that color modes supported for MIBF are black and white, and full color. You can, if required work with pre-dithered image resources, these must be converted using the supplied image conversion utility before use in the Designer, see the “Processing IJPDS and MIBF images” section in the DOC1 Programmer’s Guide for further details.

Bitmap Image Processors

MIBF output generated by DOC1GEN is converted into bitmap page images and directed to the appropriate print heads by BIPs which also control printing over a designated width of the printer’s page buffer. Specific BIPs are responsible for printing on the front and back of the paper and are designated particular color settings.

Configuration wizard settings

The MIBF Configuration wizard allows you to define settings to be assigned to your intended MIBF output device.

MIBF wizard options

Model Selection

Printer Model

Choose the type of MIBF printer hardware you wish to configure. Select the required printer from the drop down list. Note that the configuration options presented in this wizard are dependant on the model you select here. If you select the User defined option, all printer dependant restrictions are lifted.

Machine Settings

Create separate font file(s)

This setting allows you to specify the type of MIBF output to generate, as follows:
Checked
filename.MB2– contains general configuration settings.
filename.MBF*nn* – contains font resources for each BIP.
filename.MBP*nn* – contains page commands for each BIP.
Unchecked
filename.MB2– contains general configuration settings.
filename.MDB*nn* – contains page commands and font resources for each BIP.

Where *filename* is the output specified in the Publish Wizard and *nn* represents a two digit number corresponding to a specific BIP. Note that only files for used BIPs are created.

Set font memory usage (MB)

This option specifies the maximum amount of memory that can be used for fonts. The value is dependent to the hard-disk capacity of the printer. The default value for this option is -1, indicating that font memory is not restricted.

Limit memory usage (MB)

This is the maximum amount of memory specified in MB that can be allocated to the DOC1GEN process before disk caching is used.

Printstream mode

Indicates that the print stream is intended for full width printing or, half width printing.

Max page size (inch)

This is the maximum number of printer dots available in the Y direction of the page buffer for the target printer. The default is 30 inches at the specified resolution.

(continued)	**MIBF wizard options**
Full width mode settings (pixels)	Indicates page widths and nozzle settings (in pixels) for the selected printer model/printstream mode. Note these settings are preset and cannot be changed unless you have selected the *User defined* printer model option.
Half width mode settings (pixels)	Indicates page widths and nozzle settings (in pixels) for the selected printer model/printstream mode. Note these settings are preset and cannot be changed unless you have selected the *User defined* printer model option.
Text/Image/Graphic Compression	Each of these settings determine if compression is to be used in the generated output. Note that compression is not supported by the MJP20B printer.
General Settings	
Print mode	This setting is used to indicate if the datastream should be presented in *simplex* (single-sided) or *duplex* (double-sided) mode. You also select how many logical pages you want to output per side, either *1–up* or *2–up*.
Color support	This defines if the printer is to expect *Full* (CMYK) or *None* (black only).
Separate back-side color support	This setting allows you to define separate color support when printing in duplex mode. If this option has been selected you can then define if the printer is to expect *Full* (CMYK) or *None* (black only).
Text/Image/Graphics dither algorithm	Each of these settings allow you to specify the color matching process to be used with the object type. Options are: *RAST* and *RAST 4tones* – uses raster dithering. *FSB*, *FSB 4tones*, *FSB 5tones* and *FSB 7tones* – uses the Floyd Steinberg error diffusion method for dithering. *FSB* is the recommended value for image dithering. When working with text and graphics, RAST is the recommended setting. Note that the available dithering options are based on your intended MIBF printer hardware and are reflected in the associated drop down list options.
Page X offset	This values specifies the page origin *X* offset in dots. This is needed for printing outside the cue and registration mark area and is based on the top left corner of the sheet being 0,0. Printing is shifted to the right for the *X* offset.
X 2up Adjustment	This setting is for 2up printing, specifying how much the right side of the page is shifted in the x direction.

(continued)

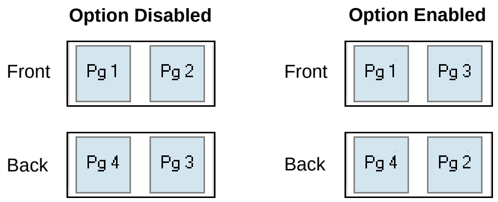
MIBF wizard options

Cue marks

These settings allow you to define the size and position of cue marks to be printed on pages produced by the application. Cue marks are used to synchronize BIPs when printing in duplex mode. These values are expressed as a number of printer dots.
You can select which BIPs are to print cue marks using the Set Cue Mark option on the BIP assignments page which is detailed later in this section.

Alternate Back/Front

This option changes the page layout in 2-up duplex printing as follows:



Place multiple publication on same sheet

This option enables different publications to be placed on the same cut sheet of paper. A new publication will not be printed on the back side of a sheet of another publication when this option is disabled.

Registration marks

These settings enable each page of the publication output to contain a print mark by the print heads individually when printing in full color mode, allowing alignment problems to be resolved. The values ar specified as a number of printer dots.

Size

This setting indicates the size of all registration marks, which is a circle set within a cross.

Page Y offset

This value specifies the page origin Y offset in dots. This is needed for printing outside the registration mark area and is based on the top left corner of the sheet being 0,0. Printing is shifted down the sheet for Y offset.

Reg mark X,Y position

These settings must be specified as a pair to indicate the offset to be used for the top left of the mark. You can set between 1 and 4 registration marks depending on the selected check boxes.

Color attributes

These settings allow you to map the fixed color definitions available to your publication design with specific CMYK values to be passed to the printer. By tuning these setting you can more closely match DOC1 standard colors to those on your printer hardware.

Black, Brown, Green … White

Enter a set of 4 integers (0-255) separated by commas to define the associated CMYK color value.

(continued) **MIBF wizard options**

BIP assignments

These settings allow you to assign specific colors to BIPs used by your printer. The information below summarizes the recommended color assignments for each available BIP.

Back side	Color	Front side	Color
BIP00	Yellow	BIP08	Yellow
BIP01	Cyan	BIP09	Cyan
BIP02	Magenta	BIP10	Magenta
BIP03	Black	BIP11	Black
BIP04	Not used	BIP12	Not used
BIP05	Not used	BIP13	Not used
BIP06	Not used	BIP14	Not used
BIP07	Not used	BIP15	Not used

Set Cue Mark — This option allows you to select which BIPs will print cue marks.

Color lookup tables (Front/Back)

These settings allow you to associate color linearization tables for use when printing text, line/boxes and images. Note that the path/file name or label entered on this page must conform to the convention required for the host operating system.
When working with different front and back color modes you can set up color lookup tables for each side of the page.

Black — Associates the linearization tables to be used when printing in black only and full color modes.

Cyan/Magenta/Yellow — Associates the linearization tables to be used when printing in full color mode.

Black generation — Associates the linearization tables to be used with the black generation RGB to CMYK color conversion functions. Used in full color mode only.

Under color removal — Associates the linearization tables to be used with the under color removal RGB to CMYK color conversion functions. Used in full color mode only

File paths

The file paths page allows for setting the path and file name for the disk caching files.

Specifying a file template

Some DOC1 output formats can create documents as individual files rather than as part of a datastream. This applies to:

- PDF output using the *Split* option,
- RTF output
- HTML for e-mail output.

In these situations each file must be assigned a unique name. This is achieved by embedding a variable in the appropriate output file name assignment using the Publish Wizard. During production DOC1 Generate will automatically resolve the variable with a unique value for each file produced.

Choose from one of following variables when making file name assignments:

- %1 – a 7 digit sequential hex number based on order of processing.
- %2 – a 7 digit sequential decimal number based on order of processing.
- %3 – the value in the *Title* field which is specified as an attribute of the publication (see “Publication attributes” on page 97). If Title is missing, %1 will be substituted.

ⓘ IT IS IMPORTANT TO ENSURE THAT THE RESULTANT FILE NAMES ARE COMPLIANT WITH NAMING CONVENTIONS OF YOUR HOST OPERATING SYSTEM.

z/OS example:

```
'G1UKJB.TSTPCOM.PDF(D%2)'
```

may produce:

```
G1UKJB.TSTPCOM.PDF(D0000001)
G1UKJB.TSTPCOM.PDF(D0000002)
G1UKJB.TSTPCOM.PDF(D0000003)
...
```

Windows example:

```
TSTPCOM%3.RTF
```

may produce:

```
TSTPCOM1012831.RTF
TSTPCOM5598179.RTF
...
```

In the example above the *Title* publication attribute has been set to customer account number.

Using resource maps

A resource map allows you to override the default names given to the font and image resources created by a publishing task (and the references used to identify the resources within an output datastream). These resources are added to the DOC1 HIP file that controls a publication in the production environment.

ⓘ BY DEFAULT, DOC1 GENERATED FONT AND IMAGE RESOURCES ARE EMBEDDED IN THE OUTPUT DATASTREAM CREATED BY A PUBLICATION. IF YOU WANT TO REFERENCE EXISTING EXTERNAL RESOURCES YOU MUST DISABLE THE *EMBED FONTS* OR *EMBED IMAGES* OPTIONS FOR THE APPROPRIATE OUTPUT DATASTREAM OBJECTS.

You may want to use a resource map to make the resource names more identifiable or if you want to use existing resources in your output datastream environment rather than those created by DOC1.

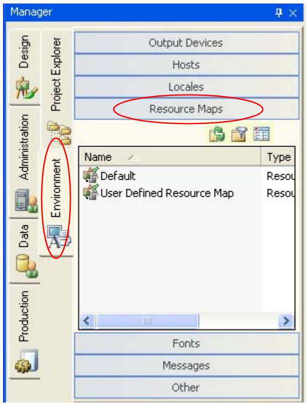
A Default resource map is always included in the repository and contains references to the fonts and images used in any publication that has been published. This cannot be edited but you can override the default mappings by creating a Custom resource map or by using the User Defined Resource Map, which contains override mappings for existing publications. Override resource maps are used in conjunction with the Default map when you publish.

▶ To create a custom resource map:

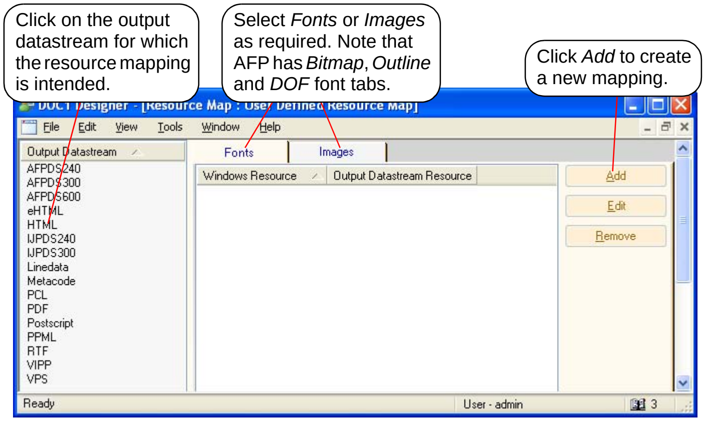
1. In the Work Center Manager select the *Environment* tab and then click the *Resource Maps* bar.

2. From the File menu select *New Resource Map* – a resource map object is placed in the navigator.

3. Enter a name for the object and press Return. The resource map editor window is opened with the object loaded.



4. Create a new mapping entry.



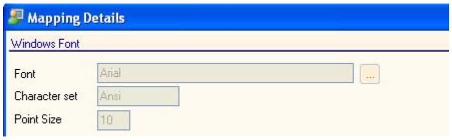
Click on the output datastream for which the resource mapping is intended.

Select *Fonts* or *Images* as required. Note that AFP has *Bitmap*, *Outline* and *DOF* font tabs.

Click *Add* to create a new mapping.

▶ **To use font mappings:** use the *Mapping Details* dialog box for *Fonts* to specify either a new resource name or a cross reference to an output datastream resource known to the repository. The resulting dialog box depends on the selected output datastream.

For **AFP bitmap** & **Metacode fonts**, each combination of font-face, point size and character set counts as an individual font that can be mapped. You should open the appropriate font folder and select the specific point size you want to map.



Mapping Details

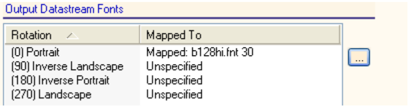
Windows Font

Font Arial

Character set Ansi

Point Size 10

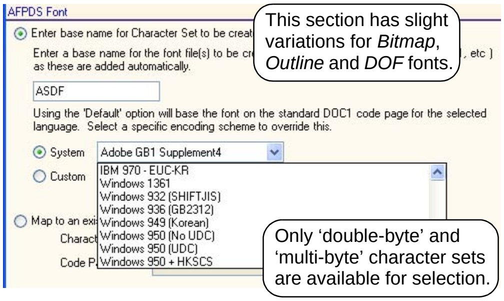
For **Metacode** fonts you can specify new names for portrait and other rotations or you can map the rotations to an existing font.



Output Datastream Fonts

Rotation	Mapped To
(0) Portrait	Mapped: b128hi.fnt 30
(90) Inverse Landscape	Unspecified
(180) Inverse Portrait	Unspecified
(270) Landscape	Unspecified

AFP font encoding can be changed from the default settings to indicate the source character set or outline font family from which the AFP font is to be generated. This option is used to change the double-byte (DBCS) and multi-byte (MBCS) associated with fonts of non western languages, such as Chinese, Japanese and Korean. As such only applicable coding schemes supporting these languages are available for selection. Single-byte characters cannot be selected.



AFPDS Font

● Enter base name for Character Set to be creat

Enter a base name for the font file(s) to be cr), etc) as these are added automatically.

ASDF

Using the 'Default' option will base the font on the standard DOC1 code page for the selected language. Select a specific encoding scheme to override this.

● System Adobe GB1 Supplement4

○ Custom

○ Map to an exi

Charact

Code P

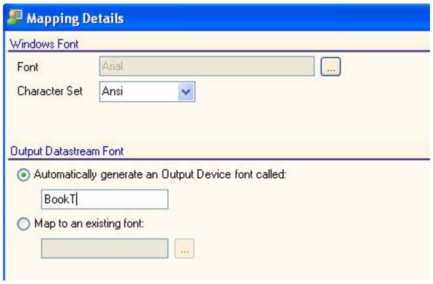
This section has slight variations for *Bitmap*, *Outline* and *DOF* fonts.

Only 'double-byte' and 'multi-byte' character sets are available for selection.

Existing **AFP bitmap and outline fonts** are a combination of up to three file types:
– a character set containing the actual font bitmaps;
– a code page that maps code points to characters;
– a coded font that references a character set and a code page.

It is possible to reference a font using a coded font, a character set/code page combination, an outline font family or a printer resident font directly. If you specify a new name for a font it will be used as the base name for all required files. Note that for bitmap and outline fonts, the name can be no more than six characters.

If you want to *Map to a … font* you should specify either a coded font or character set/code page combination, or an outline font family. Note that for AFP outline fonts you can only map to fonts that have previously been imported – see “Fonts” on page 85 for information on importing fonts.


Mapping Details

Windows Font

Font: Arial

Character Set: Ansi

Output Datastream Font

● Automatically generate an Output Device font called: BookT

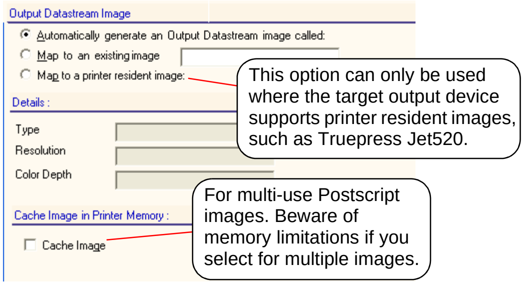
○ Map to an existing font:

For **PostScript**, **PDF** and variants all point sizes of a font-face are mapped together. You need only select the appropriate font folder and character set when mapping.

Note that if you select a double-byte font you can only map to a coded font.

▶ **To use image mappings:** from the resource map editor window, click on the *Images* tab and use the *Mapping Details* dialog box to select the image to be mapped. You can specify either a new resource name or a cross reference to an output datastream resource known to the repository.

Note that PostScript images can be cached in printer memory as required to increase performance and reduce the size of the output datastream. See“PostScript” on page 291 for further information.


Output Datastream Image

● Automatically generate an Output Datastream image called:

○ Map to an existing image

○ Map to a printer resident image:

Details :

Type

Resolution

Color Depth

Cache Image in Printer Memory :

□ Cache Image

This option can only be used where the target output device supports printer resident images, such as Truepress Jet520.

For multi-use Postscript images. Beware of memory limitations if you select for multiple images.

▶ **To delete entries from a resource map:** open the appropriate custom resource map in the resource map editor. Select the mapping to be deleted and click *Remove*.

▶ **To edit an override resource map entry:**

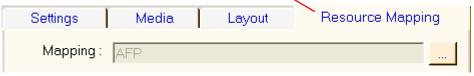
1. Double-click the resource map object you want to edit and it opens in the resource map editor.
2. Select an output datastream and the mapping that you want to change and click *Edit*.

3. Use the *Mappings Details* dialog box to specify alternate attributes for the resource mapping.

▶ To associate a resource map with an output device:

1. Open an Output Device object and select *Resource Mapping*.

   You can access an output device from the *Environment* tab, Production Job or Publish Wizard.



Select the *Resource Mapping* tab to access the *Mapping* field.

Mapping : AFP

2. Click the browse ... button on the *Mapping* field and the *Select Resource Map* dialog box is displayed.
3. Choose the required resource map and click *Open*.

Publishing and deployment

Running a publishing task in the DOC1 Work Center prepares the material required for DOC1 Generate to process a production job. You can either publish for DOC1 Generate or for the EngageOne Interactive environment.

Publishing for DOC1 Generate

A production job is made up of one or more publications that share the same input data file. You can publish the complete publication, which brings together the selected publications and creates a single HIP file containing the publication design and resources for use with DOC1 Generate. In certain circumstances you may want to publish the resources and publication design separately; for example, in a large scale production environment this could save time if you just need to update the resources. You can produce any number of separate resource HIP files for use with Generate.

Publishing for Interactive

In the EngageOne Interactive environment, a template package is created from one publication. This comprises a manifest that describes all the files that make up the template. See “Publishing for EngageOne” on page 218 for more information.

Publishing Active Content

Active Content can be published independently and then used dynamically to update one or more publications that are already in production. This may be useful where you have several publication designs that reference the same Active Content object and where frequent updates to the content are required. However, you should exercise care when using this option as changes made to Active Content may have a significant impact on production jobs. Note that when publishing for DOC1 Generate, the use of extra fonts or images will require the publication resources to be regenerated. This includes the use of additional point sizes of fonts already used and new keyed image maps.

About publishing

Publishing always works with issued versions of Active Content or publications and the other files they reference – i.e. draft versions of files will not be included. However, you can use the *Publish Current Draft* task to issue files automatically prior to publishing if required. Specifically, the last saved version of the Active Content or publication is issued prior to publishing along with all the documents and other objects it references providing they are marked as private for the publication.

The *Publish Issued Version* task will ignore any draft versions of the publication, Active Content, or referenced resources that have not yet been issued.

Before you can complete a publishing task all data fields/records that have been used in the publications, or Active Content must be linked with actual data elements that will appear in the input file to be used by Generate. These references are resolved by creating, updating or referencing a data map.

ⓘ RESOURCES ARE AUTOMATICALLY PRIVATE IF THEY ARE CREATED WITHIN ANOTHER OBJECT (E.G. AN ACTIVE CONTENT OBJECT CREATED WHILE EDITING A SPECIFIC DOCUMENT) ALTHOUGH THEY CAN BE CHANGED TO PUBLIC AS REQUIRED. RESOURCES CREATED INDEPENDENTLY ARE AUTOMATICALLY PUBLIC. REFER TO "PRIVATE/PUBLIC RESOURCES" ON PAGE 373 FOR MORE INFORMATION ABOUT PUBLIC VS. PRIVATE ASSIGNMENTS.

Publications

When you run a complete publishing task for a publication you will need to specify:

- one or more **output device** objects that identify the type of datastream to be produced
- a **host** object that provides information about the operating system on which DOC1 Generate will process the job and any settings required to communicate with the production machine when deploying the HIP file.

Pre-configured output device and host objects are provided with the DOC1 Work Center that reflect the most common requirements of the configurations indicated by your DOC1 license. You will probably want to make some modifications to these to provide custom settings before running a publishing task for the first time.

ⓘ FOR DETAILED INFORMATION ABOUT CUSTOMIZING OUTPUT DEVICE OBJECTS SEE "OUTPUT, MEDIA AND RESOURCES" ON PAGE 274.

You can retain information about the publishing options you use in a Production Job object. If required, production jobs can be created in advance and can then be made available to users responsible for publishing.

When the publishing task completes, the HIP file is created and automatically deployed to the location indicated in the assigned Host object. Normally this will be a file location on the actual server where Generate will run but if you do not have direct connectivity with the intended production machine the host object can indicate a local directory in which the HIP will be created.

Separate design and resources

When publishing the design only, a single design HIP file will be produced – with no resource information. Resources must be published separately. You can either publish all the required output devices to a single resource HIP file or publish them separately in multiple resource HIP files.

Note that when publishing the design and resources separately you will need to make the resource HIP files and output file names known to DOC1 Generate in the OPS file; see “Using segmented resources” on page 349.

Active Content

If you reference publishable Active Content in your publication you must make sure that the Active Content has already been published. For details of how to specify the location of the Active Content HIP files in DOC1GEN see “Using segmented resources” on page 349.

About data mapping

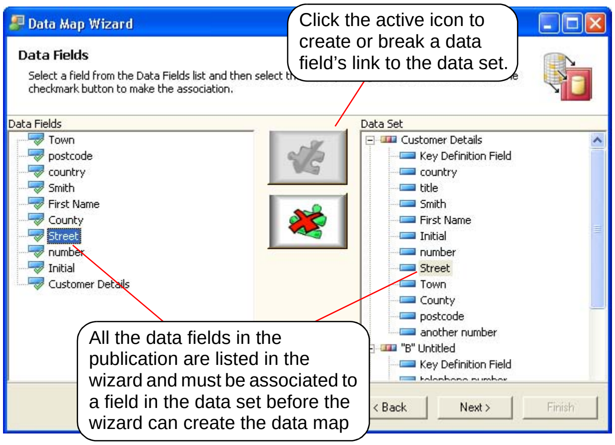
A data map links the data aliases (fields and records) you have used in your document designs to actual data elements in the input file to be used by Generate. The definition of the input file is stored in a data format and it follows that a data format must have been created for your intended input data before you can publish.

As part of a publishing task you will need to associate each type of publication with a data map or create a new data map if required. Where your publication design contains data aliases that are not resolved in the selected data map, a dialog box is displayed listing the unmapped aliases and then the data map wizard will be initiated to allow you to provide the required links to the data format.

> ⓘ For details about creating a data format and exporting a data map see “Creating a data map” on page 264.
> For details about working with interactive data for EngageOne see “Generating a dictionary and data map for IDE” on page 210.

The links you specify using the wizard are stored in a data map object and this can be reused in subsequent tasks and can be shared by other publications if appropriate.

You can create a data map in advance (along with a data dictionary to be used in document designs) by using the *Generate Dictionary* option when editing a data format.



Data Map Wizard

Data Fields

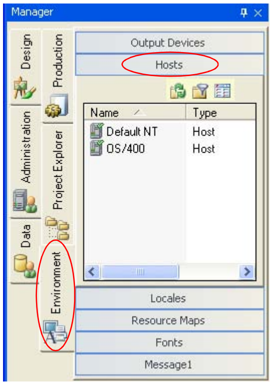
Select a field from the Data Fields list and then select th checkmark button to make the association.

Click the active icon to create or break a data field’s link to the data set.

All the data fields in the publication are listed in the wizard and must be associated to a field in the data set before the wizard can create the data map

Creating a host object

Host objects are used to specify the type of operating system on which DOC1 Generate will run plus the settings to be used when deploying HIP files to the host machine. Note that host objects are not required when publishing for EngageOne Interactive.



Name	Type
Default NT	Host
OS/400	Host

▶ To create a host object:

1. In the Work Center Manager select the *Environment* tab and then click the *Hosts* bar.
2. From the File menu select *New Host*.
3. The new host object is loaded in the workspace ready for editing.

▶ To edit an existing host: double-click on the object in the relevant Work Center Manager view.

Host object settings

Type

Indicate the type of *Operating system* in use on the host.

Connection

This section indicates how the DOC1 Work Center will communicate with the host system when deploying the HIP file as part of a publishing task.
If you do not have direct connectivity with the host or if the host is the local machine or a computer directly available on the LAN you should select the *Local* option. Otherwise select *FTP* to indicate the protocol to be used to communicate with the host system.
The *Address* used for FTP hosts can be specified either as a computer name or as a TCP/IP address.
The remaining options in this section allow you to specify the account and security settings to be used when communicating with the host.

Location

This allows you to indicate the location on the host system where HIP files will be deployed. Always use the notation appropriate for the intended host system:
for Windows or UNIX specify a folder using UNC syntax or equivalent;
for z/OS specify a fully qualified dataset name and if required indicate that the dataset will be partitioned;
for OS/400 use the libname/filename convention;
for OpenVMS specify a path including the bracket notation.

(continued) **Host object settings**

The *Hip Filename* option allows you to provide a specific name for the hip file. If no name is specified the hip will be given the name of the production job. The following parameters can be used in the creation of the hip filename:
%B – DOC1 build version
%D – deployment date, the date format is yyyymmdd
%I – production job short name (first 8 characters)
%J – production job name

%M – data map name
%N – publication name; for multiple publications the name of the first one is used
%P – project name; for multiple projects, the first publication's project is used
%R – revision of the publication; for multiple publications the highest revision is used
%T – deployment time; the time format is hhmmss
%U – the full name of the user who created the hip
Example:
%D_%I = 20040215_CurrentA.hip

Licence

Enter the DOC1 Generate *Company Name* and *Licence Code* for the production platform to which this object relates. These will have been provided to you in a DOC1 keycode report. Note that both settings are case sensitive and should include any punctuation included on the keycode report.

Specifying files for Generate

At publishing time you will need to indicate the files that will provide input to a production run and those that will receive the output created by the job.

Normally you will want to assign specific references for such files, i.e. you will specify the location and file name where DOC1 Generate will look for them or create them.

If required, you can choose the *Set input/output on host* option to allow the actual file references to be specified in an Override Production Settings file (OPS) that is passed to Generate when a production job is actually started. This allows you to amend file references for individual production runs. You will need to take a note of the *Name* assigned to each file by the publication task as you will need to specify this as part of the OPS format.

ⓘ Information about creating an OPS file for use with Generate can be found in "Running DOC1 Generate" on page 348.

Additionally, where the intended host system is z/OS you can specify file references as DD labels that can then be resolved to actual dataset names in the JCL used to initiate Generate.

Where you are using specific file references you should always use the notation appropriate for the intended host system. For example, if Generate is going to run on a Windows or UNIX platform you might use a server, folder and file name reference; for z/OS you should use dataset/member references; and for OS/400 you should use the libname/filename(member) convention.

If you specify only a file name (i.e. no other server/path references) Generate will normally search for the file in the location from where it was started.

Note that the HIP file that is created by the publishing process is always passed to Generate as part of the batch start-up command.

Production job file references

Input

Input files

These can comprise:

- the primary input data file for the production job that contains the variable information presented in the documents created.
- one or more references to resource hips generated when you publish resources separately from the publication.

Lookup tables

If a publication uses the table lookup feature to map text references to alternative strings you will need to specify one or more files that contain the lookup information. Refer to “Lookup table” on page 153 for information about the expected structure of a lookup table.
Table references the File Alias used in the lookup table object in the publication.
File is the location of the lookup table on disk.
Terminator enables a character to be used to designate the end of a record entry when using a lookup table with z/OS or OS/400.
Code page enables a system or custom code page to be associated with the text in a lookup table.

External key maps

If a publication uses keyed images that are not already in the Work Center, then you must specify the process that will supply them. This can be either a third party file that provides the key map along with image details, or a user exit that performs the mapping and supplies the actual image. A third party file is just specified by its filename; a user exit is specified by using the format `USEREXIT(`*`function`*`)`, where function is the registered user exit name. For more information see the “User Exit” section in the DOC1 Programmer’s Guide.

If you are using a third party file to identify the images, then this file must be available on the host machine and the images themselves must be available to the printer (or on the host machine if you are generating PDF or IJPDS). Images provided by a user exit must similarly be made available, for example, retrieved from a database.

(continued)

Production job file references

The *Mode* option specifies how the key map is read at run time:

Cached keys in memory will copy the key map into memory, making access much faster. This is the default setting.

Run time lookup will access the key map externally each time. This does not use any extra memory, but will not be as fast. You may want to use this option if there are a great number of entries in the map and memory is limited.

Output

Output files — These are the files that will receive the output datastream from the job; i.e. the composed pages ready to be sent to the target output device and any associated work files. The number of files expected depends on the number of output devices specified for the production job and if outsorting is being used. For information about the structure of the output file refer to the information for your intended output datastream in “Output, media and resources” on page 274.

Journals — If a publication writes data to one or more journal objects you will need to provide a file reference for each. You can also assign a specific code page to a journal.

Running a publishing task

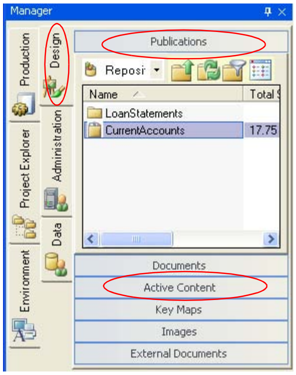
The publishing tasks run a wizard that steps you through the options required to prepare a production job. This can be to publish the complete publication, to publish the publication design or resources separately, or to publish Active Content. Note that before you can publish Active Content you have to enable it for publishing – see “Using Active Content to update publications in production” on page 233.

When publishing publications you can identify the intended host system and output devices, specify the file references to be used by Generate on the host system and set any other run-time options relevant to the job.

You can select an existing production job object to load the settings it contains into the wizard. If you are entering new settings or updating those loaded from an existing object you will be able to save them to a production job when the wizard finishes.

▶ **To run a publishing task:**

1. In the Work Center Manager window select the *Design* tab and then either the *Publications* bar
   – or –
   the *Active Content* bar.



2. Highlight the publications or Active Content to be published. Use shift+click for multiple publications.
3. From the *Task* menu select *Publish Issued Version* if the publications are already issued
   – or –
   select *Publish Current Draft* task if you have not yet issued the publications.

   The Publish Wizard is initiated.
4. Load settings from a production job if required.

   Click *Select* to display the *Select Production Job* dialog box. You can either create a new production job or choose an existing one from the repository.
5. Complete data mapping if required.

   Each publication must be associated with a data map. Double-click a publication to assign, amend or create a data map for it
   – or –
   click *Assign Data Map* to associate a data map with a publication or create one from a data format using the Data Mapping Wizard.

   You will be asked to update the selected data map if it does not contain references for all data links within the publication.

   See “About data mapping” on page 331 for details.

6. Identify the type of output datastreams to be produced by the job by selecting one or more Output Device objects.

   See “Output, media and resources” on page 274 for details.

7. Use the Input/Output page to identify the type of host system intended to run the job and to specify references to all files be used by Generate on the host.

   See “Creating a host object” on page 332 and “Specifying files for Generate” on page 333 for details.

8. Select other publishing options required and click *Finish*.

   If all publication options are valid, a HIP file is created and deployed to the location specified in the assigned Host object.

   > ⓘ If a mandatory option in the Publish Wizard has not been resolved the corresponding page title is displayed in Red and the *Finish* button is disabled.

   If you have updated the settings from an existing production job object you will be prompted to save them.

Publish Wizard options

Publish Options

Batch/On demand

Publications
This will publish the files required by your publication to run Generate on the host system and to print/present the resulting output datastream on your intended output devices. You can publish all the publication design and resources at the same time, or you can choose to publish them separately Choose from:
Complete publication produces a single HIP file that supplies all the publication design and resources.
Publication design only produces a single HIP file containing the rules for constructing the publication.
Publication resources only produces a single HIP with all the resources (fonts, output device settings, images and overlays) to support the publications you have selected. You can produce any number of resource HIP files for a publication. They can be for any combination of output devices required for the publication – one, some or all. One or more can then be used when running Generate.
Note that when publishing design and resources separately the resource HIP files must be specified as input files to Generate. See the “ResourceHIP” option in the <Input> section of the OPS file on page 353 for more information.

(continued)

Publish Wizard options

Active Content
This will publish an individual HIP file for the Active Content selected. If you want to publish more than one Active Content object click *Next*. Use the ⊞ icon and browse through the folders and publications in the repository using the window provided. Any publishable Active Content objects found within a selected publication are added to the publishing task, as well as:

- any publishable Active Content objects referenced within other Active Content,
- publishable Active Content referenced by keyed Active Content maps.

You can use the ⊟ icon to remove Active Content objects.

Click *Finish* to activate the publishing task. Use the window provided to browse to a destination folder, or create one for saving the HIP files.

Interactive

Publications
This will publish an EngageOne template package created from a single publication. This is a combination of presentation elements, logic, interactive data placeholders and a manifest describing the attributes and contents of the template. See “Publishing for EngageOne” on page 218 for more information. Note that templates do not contain any font and image resources for target output datastreams. Such resources are placed within independent files that must be created and deployed in conjunction with requests from the EngageOne server. See “Publishing resources for EngageOne” on page 218.

Active Content
This will publish one or more Active Content objects in a single zip file for the EngageOne Interactive environment. See “Using Active Content with EngageOne” on page 234.

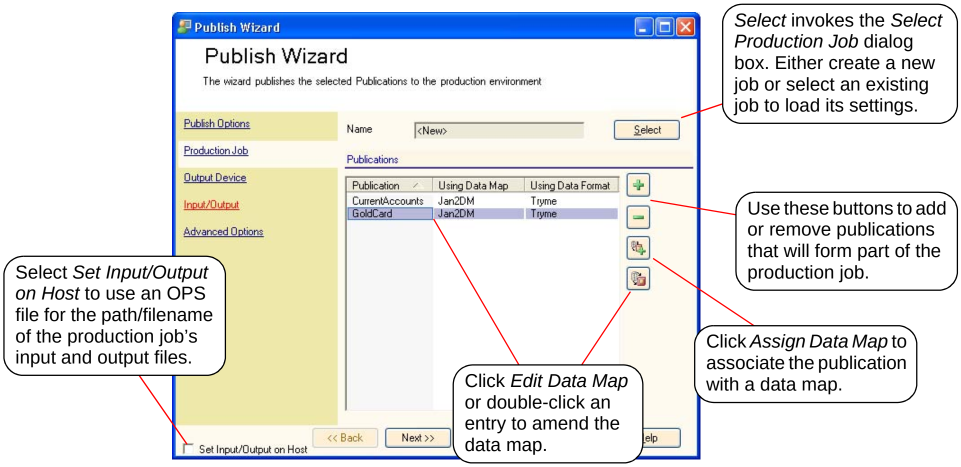
If you want to publish more than one Active Content object click *Next* and follow the instructions above in “Batch/On demand” for selecting the required Active Content.

Click *Finish* to activate the publishing task. Use the window provided to name and save the zip file to the location required.

(continued) Publish Wizard options

Production job

Note that the options available on this page depend on whether you are publishing a complete publication, the design only or the resources only.



Publish Wizard

The wizard publishes the selected Publications to the production environment

Name <New>

Publication	Using Data Map	Using Data Format
CurrentAccounts	Jan2DM	Tryme
GoldCard	Jan2DM	Tryme

□ Set Input/Output on Host

Select invokes the *Select Production Job* dialog box. Either create a new job or select an existing job to load its settings.

Use these buttons to add or remove publications that will form part of the production job.

Click *Assign Data Map* to associate the publication with a data map.

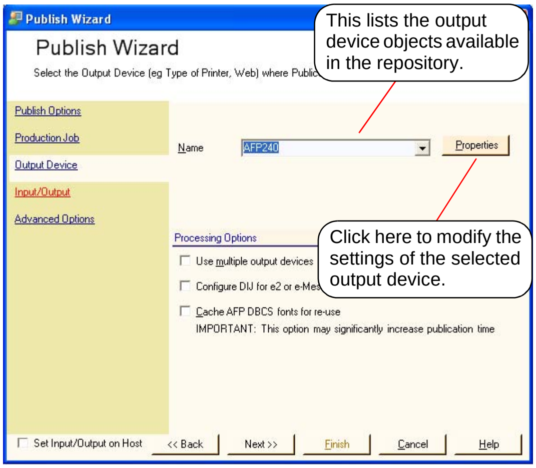
Click *Edit Data Map* or double-click an entry to amend the data map.

Select *Set Input/Output on Host* to use an OPS file for the path/filename of the production job's input and output files.

(continued) **Publish Wizard options**

Output Device

Process options



Publish Wizard

Select the Output Device (eg Type of Printer, Web) where Public

Name: AFP240

Processing Options

□ Use multiple output devices

□ Configure DIJ for e2 or e-Mes

□ Cache AFP DBCS fonts for re-use

IMPORTANT: This option may significantly increase publication time

□ Set Input/Output on Host

This lists the output device objects available in the repository.

Click here to modify the settings of the selected output device.

Note that all the options in this section are made available depending on the user profile being used.

Use multiple output devices

If selected multiple output datastreams are created in relation to the same set of input data.

Configure DIJ …

If selected a DOC1 Interchange Journal (DIJ) will be created when the job is processed by Generate. This is required when the output datastream is to be used with the e2 Vault, Present and Pay, and e-Messaging products. When this option is selected an additional page is provided in the wizard to allow you to specify the file to receive the DIJ and to provide related settings.

Cache AFP DBCS fonts for re-use

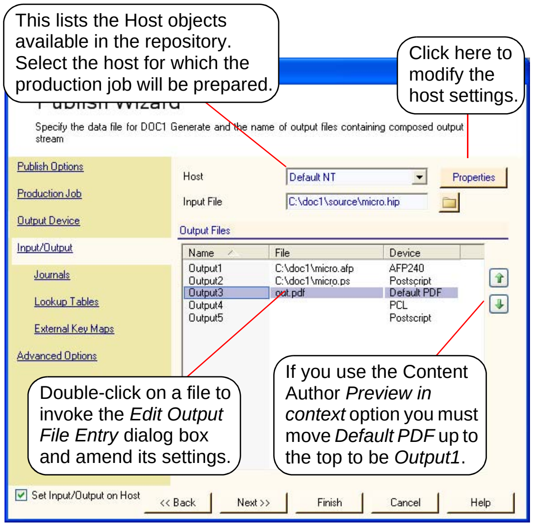
If selected AFP DBCS fonts are automatically stored in your local file cache. Selecting this option can result in an increase in publication time, however these fonts will be re-used by subsequent publications.

(continued) **Publish Wizard options**

Input/Output

These pages allow you to indicate the files that will provide input to a production run and those that will receive the output created by the job. Refer to “Specifying files for Generate” on page 333 for details.

Note that the *Input File* browse button is disabled if a non-Windows host is selected.



This lists the Host objects available in the repository. Select the host for which the production job will be prepared.

Click here to modify the host settings.

Double-click on a file to invoke the *Edit Output File Entry* dialog box and amend its settings.

If you use the Content Author *Preview in context* option you must move *Default PDF* up to the top to be *Output1*.

Specify the data file for DOC1 Generate and the name of output files containing composed output stream

Host: Default NT

Input File: C:\doc1\source\micro.hip

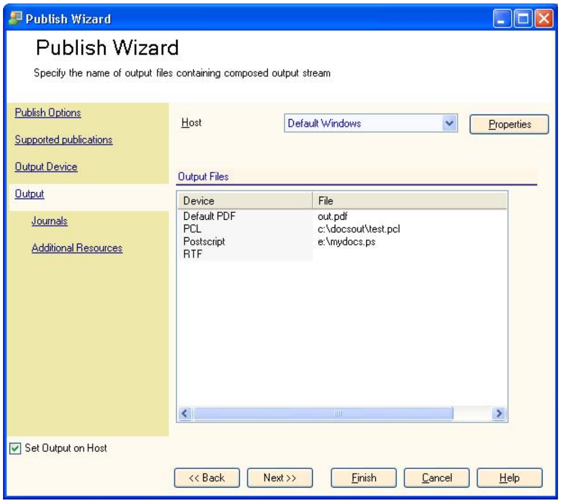
Output Files

Name	File	Device
Output1	C:\doc1\micro.afp	AFP240
Output2	C:\doc1\micro.ps	Postscript
Output3	out.pdf	Default PDF
Output4		PCL
Output5		Postscript

☑ Set Input/Output on Host

Output

These pages are used when publishing resources separately and allow you to indicate the files that will receive the output created by the job. Refer to “Specifying files for Generate” on page 333 for details.



Publish Wizard

Specify the name of output files containing composed output stream

Host: Default Windows

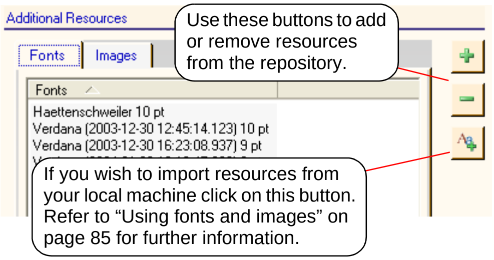
Output Files

Device	File
Default PDF	out.pdf
PCL	c:\docsout\test.pcl
Postscript	e:\mydocs.ps
RTF	

☑ Set Output on Host

(continued) **Publish Wizard options**

Additional Resources



Haettenschweiler 10 pt
Verdana (2003-12-30 12:45:14.123) 10 pt
Verdana (2003-12-30 16:23:08.937) 9 pt

Use these buttons to add or remove resources from the repository.

If you wish to import resources from your local machine click on this button. Refer to “Using fonts and images” on page 85 for further information.

Additional font and image resources can be included in your publications HIP file. These can either be added from the repository or imported from your local machine.

This option can be used to add extra resources referenced by RTF fields included in your input data. Refer to “Including RTF fields in input data” on page 182 for further information.

Note that when including native printer fonts as additional resources you must ensure these match your intended output datastream type.

In an EngageOne Interactive environment you can specify additional fonts that will be made available to a correspondent when working with editable text in the front-office environment.

Advanced Options

These pages allow you to fine-tune the DOC1 Generate run-time environment.

Publications to process

Select *Range of publications* if you want DOC1 Generate to process only a subset of the publication data sets available in the input data file. You may want to do this if you need to rerun portions of a production job without creating a new input data file. Use the entry field to indicate the sequential numbers of the publication data sets to be processed. For example:
`27,280,674` – specific publications
`100-1000` – publications between 100 and 1000
`1000+` – all publications after the first 1000
`366,500-1000,2000+` – combinations.

Checkpoint progress

Use these options to indicate when checkpoints are to be created by DOC1 Generate. Checkpoints output a message at user-defined intervals indicating which publication data set is currently processing. This information is used if you want to restart processing after a failure.

If you want to restart processing from the last checkpoint position you will need to specify a *checkpoint file*. You must also specify *#restart* when restarting Generate, either in the OPS file (see “Restart” on page 358) or on the command line (see page 363 to page 366).

(continued) **Publish Wizard options**

Use the *Every* **n** *publications* setting to indicate the frequency with which checkpoints are issued. This is the number of publication data sets to be processed from the input data file before the next checkpoint message is raised. If required, you can also use *Starting after* setting to indicate that checkpointing messages should not be issued until the specified number of publication data sets have been processed.

Error Handling

On error with publication data set

This setting relates to the handling of keyed record data only. Use it to indicate what action should be taken if DOC1 Generate encounters a publication data set that cannot be processed. This may occur in the following circumstances:
1. A record is defined as Mandatory but is missing in a publication data set.
2. An undefined record key is encountered.
3. A child record of a repeating group is present without its parent record.

Options are:
Abort job – no further publication data sets are processed and any output files that have been created by the job are deleted (if this is permitted by the host operating system).
Skip publication – the publication data set in error is ignored and no output is created from it. Processing continues with the next publications data set in the input file.
Record data and skip publication – as above but the publication data set is also copied to the file indicated by the *Data record file* field before proceeding. Note that this file specification must conform to the same naming conventions as all other Generate I/O files; refer to “Specifying files for Generate” on page 333 for details.

When shape is positioned off page

These options indicate what action DOC1 Generate should take if a graphic object (including text boxes) have offset values that mean they are positioned all or in part outside the active logical page area. Different actions are possible for graphics that are placed using **constant** values and those that use **variable** data (i.e. when using any value object other than a constant to provide an offset).

Options are:
Abort – Generate aborts immediately. Any output files that have been created by the job are deleted (if this is permitted by the host operating system). This is the default for objects placed using variable data.

Warn – a warning message is issued for each object that is found to be positioned outside the logical page area. Processing of the job continues as normal. The off page object is included in the output datastream; the effect of this in the printer/browser environment will depend on the device type.
Ignore – as above but no warning message is issued. This is the default for objects placed using constant values.

(continued)	**Publish Wizard options**
Suppress messages	Use this setting to indicate any DOC1 Generate diagnostic messages that you do not want to be issued for this production job. You can either indicate the category of messages to be suppressed or use the *Only these messages* option to indicate specific messages that you do not want to be reported. When using this option you must specify only the identifier that appears before each message when issued.

Memory Handling

The default setting of *Allocate memory as required* allows DOC1 Generate to call for system memory in the amounts most suitable for its processing needs. If you need control over such allocations select the *Pre-allocate memory at start up* option and then use the *Allocate memory blocks of size* setting to specify the amount of additional memory to be reserved each time a call is made. You can also use the *Start up with* setting to indicate an initial memory allocation prior to any additional blocks being called.

If you need to set a maximum limit for the total amount of concurrent memory that can be allocated by Generate, select the *Abort if memory exceeds* option and specify the limit in the entry field.

By default, Generate will attempt to hold all the pages of output datastream created for an individual publication in system memory until it is complete. This means that Generate can perform output functions as efficiently as possible. You can use the *Limit composed pages* option to specify an upper limit to the amount of memory to be allocated for this purpose. If this option is enabled a default of 4mb will be used as the limit although you can adjust this to any value above 512k. When the memory requirement of a publication exceeds this limit the existing pages in memory will be output to a local disk file which is used as temporary work space. When the publication is ready for output the pages in the work space will be retrieved and merged with those still in memory. By default, Generate will create a temporary file in the current location for this purpose although you can indicate a specific path or file name if required. Note that this file specification must conform to the same naming conventions as all other Generate I/O files; refer to “Specifying files for Generate” on page 333 for details.

Note that all system memory settings are, by default, expressed as kilobytes. You can alternatively use the M suffix to indicate a number of Megabytes if required. For example:
64 = 64 kilobytes
100M = 100 megabytes
128K = 128 kilobytes

(continued)

Publish Wizard options

Diagnostics

Issue report of memory usage

If this option is enabled DOC1 Generate will issue a summary of the amount of memory allocated by the job on completion. You may want to use this option during testing to optimize memory handling settings.

Log file

This option is used to specify the file that will receive any error or warning messages issued by DOC1 Generate. The filename and path must be in the required format for the operating system.

Produce trace information

This provides a method of tracing publications that cause errors in Generate or Preview. A tree structure representing the path that Generate took that caused the error is displayed. This enables the error to be tracked down to a particular object in the publication design. Note that this can also be specified in the OPS file. See also “Debugging publications” on page 350 for more information.

Trace output file

Specifies a file for the output from the trace. The filename and path must be in the required format for the operating system.
If no file is specified the output is sent to the standard output medium of the Generate host environment (e.g. command prompt window, system log, etc.)

Memory limit

This is the maximum amount of memory in bytes that can be used for buffering trace information. A value of ‘0’ means no limit.

Output file codepage

This allows you to change the code page for the output file from the default host code page to one that is more suitable.

Verbose trace information

Set this to include GUID information and the internal names of the opcodes that are being executed.

Messages

If your publication design contains messages created using the Message1 or Content Author products, use this page to customize the production side of such message application environments. For more information see “Interfacing with message applications” on page 191.

Messages

Click the ➕ icon to display the Add Message dialog box. You can search for one or more message files (.HIM) that are to be used with the publication by using the Browse button.

Audit trail file

Specify the file to be used for audit information for each message that is included in the published documents.

Mandatory not placed

Select the action you want DOC1 Generate to take when a mandatory message cannot be included in a document for which it was intended.
Stop – the default is to stop processing
Continue – ignore the error and carry on
Warn – issue a warning and continue.

(continued)	**Publish Wizard options**
Mandatory message error	Select the action you want DOC1 Generate to take when a mandatory message is found to have unresolved links – typically when a font used by the message is not included in the resource pack or when a data field used in a message has not been mapped. *Stop* – the default is to stop processing *Continue* – ignore the error and carry on.
Optional message error	As above but specifies the action to be taken for non-mandatory messages.
Use HIP font mapping	Select this to ensure that any mappings that have been applied to fonts used in a publication in the Designer (i.e. are in the HIP file) will also be applied to the same fonts used in messages created in Content Author or Message1.
Server Mode	
	This section is used if you want your publication to be processed using the Server Mode method of running DOC1GEN under UNIX and Windows. For more details on Server Mode, see the DOC1 Programmer's Guide.
Run DOC1 Generate in Server Mode	Select this option to activate Server Mode and enable its related settings. *Abort on failure* defines the behavior of DOC1GEN if a print job fails. Selecting this option will abort Server Mode, otherwise Server Mode attempts to discard the failing print job and await further commands.
Command Queue	Server Mode uses a command queue to receive commands and application data. The command queue can be specified either as a named pipe or a TCP/IP socket. The format for *named pipe* under NT should be `\\.\pipe\cmdqueue` and under UNIX it should follow a path/filename format such as `/temp/cmdqueue`. For *TCP/IP* the format for *Socket* is an IP address such as `10.133.54.202` and for *Host* it would be a host name, such as `spa02`. Both use the *Port* field which is a number corresponding to the required port.
System Commands	These commands can be specified to be executed before Server Mode has processed application data, if it completes successfully or if it fails to complete successfully. They can be any commands that can be passed to the appropriate operating system from the command line.
Custom	
	This page allows you to enter any special settings that may be required as part of problem resolution. Only enter values here at the request of DOC1 customer support.

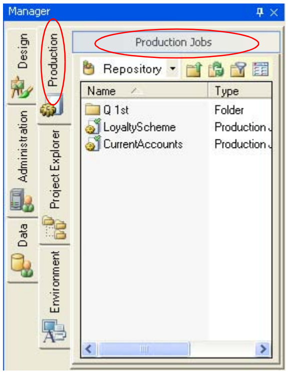
Working with production jobs

A production job object holds settings to be used with publishing tasks in the DOC1 Work Center. You can create a production job at the time when the Publish task is run to gather the options you specify. Alternatively you can create a new object in the Work Center for future use.

▶ To create a production job object in advance:

1. In the Work Center Manager window select the *Production* tab and then click the *Production Jobs* bar.
2. From the File menu select *New Production Job*.
3. Enter a name for the production job in the navigator and open it.
4. Configure the Production Job, Output Device, Input/Output and Advanced Options settings as necessary.

   The settings that you can use are identical to those available when you use a publish task. Refer to "Publish Wizard options" on page 337 for details.



Name	Type
Q 1st	Folder
LoyaltyScheme	Production
CurrentAccounts	Production

▶ To edit an existing production job: double-click on the object in a relevant Work Center Manager view.

Running DOC1 Generate

DOC1 Generate is the batch program that processes production jobs on your chosen host system. Generate reads information about the job requirements from a DOC1 HIP file, merges the input data file it receives with your publication designs and produces output datastreams ready for printing or presenting on your intended output devices.

> ⓘ On some supported platforms memory resident versions of DOC1 Generate are available (Server Mode and Started Task). These allow a DOC1 production job environment to remain loaded and for batches of input data passed to the defined channel to be processed dynamically. You will need to configure the environment before running in these modes and the launch method will differ from that described in this section. The DOC1 Programmer's Guide contains full details of these methods.

The program typically has the name DOC1GEN on all platforms. You will need to run the program from the batch environment appropriate to your host system and production process: command line, script, JCL, etc.

The name and location of the HIP file that is to control a job is specified as a parameter to the DOC1GEN program when it is started. Normally, all other file references will have been specified in the production job settings that were used when the job was published in the DOC1 Work Center. If required however, you can create an Override Production Settings file (OPS) in which you can specify additional/alternative file references and other settings. Where used an OPS file is specified as a second parameter to the DOC1GEN start-up command.

You will need to ensure that all input files referenced in the productions settings and the OPS file are available at the locations specified when DOC1GEN is started. For jobs running under z/OS indirect file references using DD names are typically used in which case the start up JCL will need to include DD cards with these labels that indicate the actual datasets.

Output files will be created with the file location/names specified in the production settings and/or the OPS file. You will need to ensure that suitable disk space is available to receive the output at the defined location. Under z/OS you will need to allocate suitable datasets either in advance or as part of start-up JCL if DD references are used.

Using segmented resources

Where you are using independently published (segmented) resources or Active Content you should be aware of the following:

- DOC1 Generate needs to know where to find the segmented resources as they are not present in the design HIP file specified on the command line. The resource HIP or HIPs necessary for a job must be specified in the OPS file using the `<Input>` section `ResourceHIP` and `ActiveContentLocation` keywords. One or more `ResourceHIP` keywords can be used, allowing device specific resources to be kept separate if desired. Segmented resources must be in the same repository as the publication design.
- DOC1 Generate has no control over resource versions when working in this way. It is the user's responsibility to ensure that all HIPs are compatible and contain the appropriate resources.
- Care must be taken if resource HIPs specified by the `ResourceHIP` keyword contain conflicting device settings for the same device type. In this case the settings used by the last loaded HIP will be used, possibly resulting in unexpected output.
- The `<Output> Name` keyword is specified differently when resources are published independently. See the <Output> section of the OPS file on page 355.

Code page support

In order to read input data and configuration settings specified on your production system DOC1GEN needs to be able to convert the data it receives into the Unicode format it expects internally. To do this it uses a range of code page tables that define the required translations. For most Western applications these tables are contained within DOC1GEN itself and you need to take no specific action in the production environment. Due to their potentially large size, code page tables for non-Western applications are stored in a separate Extended Code Page (ECP) file and you will need to ensure this is made available to DOC1GEN by referencing it when starting the program.

Return codes

DOC1GEN always returns 0 (zero) for successful completion or where warning messages (only) have been issued. Return code 16 is issued where a failure has occurred – i.e. where an abort message has been issued.

Messages

Messages issued by Generate are always written to the standard output medium for the system on which the program is running. A list of possible messages and their explanations can be found on the DOC1 Support Net web site at www.doc1supportnet.com.

Legacy support

The DOC1 Series 5 production engine cannot process jobs created in a pre-Series 5 environment without modification. However, you can use Generate to automatically launch the Series 4 production engine (known as EMFE). To do this you need to call the DOC1GEN program with an OPS file (see below for details) that indicates the location of the DOC1EMFE program and the relevant EMFE initialization file. You should refer to your Series 4 user documentation for details of the parameters and files that are required when using this method.

Debugging publications

When an error occurs in the logic of your publication design, you can use the debugging option in Generate or Preview to trace the problem. Processing stops on the first publication that causes the error and a tree structure is generated that represents the path the program took when the error occurred. This can help in tracing the problem to the relevant object in the publication design. The tree structure shows logic map labels so it is advisable to ensure that your design is fully labelled.

Associated data is also shown, although this may include data from previous publications so you must be aware of which data the publication actually uses. Information on the internal opcodes being executed can also be shown if required. The debug information is output either to a file or to the standard output device.

Debug options can be specified both in the publishing task (see “Diagnostics” on page 345) and in the OPS file (see “Trace” on page 355). The *TraceLevel* option in the OPS file can be used to switch debugging mode on or off each time you run Generate.

OPS file

An override production settings file allows you to specify supplements and alternatives to some of the job settings that were used when publishing a publication design.

The use of an OPS file is optional unless you have not specified all file references and license data in the publishing task.

An OPS is a text file. Options are coded as keywords and associated parameters within several distinct sections. Sections must be introduced with the relevant name within angle brackets, for instance: `<Journal>`. If you want to include comments in the OPS file, prefix the comment line with a semicolon character.

No sections or keywords are compulsory and you should code only those options that suit your requirements. All missing options are assumed to have been specified as part of the publishing task.

All file references can include both path and file name as required. Ensure that you code all such references in a format suitable to the operating system under which you are running the production engine.

OPS file format

Syntax:

```
<Generate>
ProgramLocation=Filename
EMFE=Filename
INI=Filename
ServerMode={True|False}

<Input>
DataInput=Filename
MessageLib=Filename
MessagesFile=Filename
ResourceHIP=Filename
ActiveContentLocation=Path

<LicenseInfo>
CustomerName=String
Keycode=Keycode

<Journal>
Name =Filename
Name =Filename
...

<LookupTable>
Name =Filename
Name =Filename
...

<KeyMap>
Name =Filename
Name =Filename
...
Mode ={Cached|Runtime}

<DIJ>
Name =Filename
...

<Output>
Name =Filename
Name =Filename
...
MessageAuditTrail=Filename
Name
```

(continued) **OPS file format**

```
<trace>
Outputfile=Filename
TraceLevel={off|default|verbose}
outputcodepage={UTF8|default}
memlimit=Memory
publication=Number

<Messages>
MandatoryNotPlaced=Stop|Continue|Warn
MandatoryMessageError=Stop|Continue
OptionalMessageError=Stop|Continue
CampaignDate=String
Cycle=String
MessageProcessing={Yes|No}
NoMessages=Stop|Warn

<Server>
CommandQueue= {QueueID|PIPE:QueueID| SOCK:address:port|HOST:hostname:port}
Commandnn=CommandString
CommandBefore=CommandString
CommandOK=CommandString
CommandFail=CommandString
CommandEnd=CommandString
AbortOnFail= {True|False}
...

<Advanced>
ErrorFile=Filename
LogFile=Filename
Checkpointfile=Filename
CPconsole= {0|1}
ConstantShapeOffPage=Abort|Warn|Ignore
DynamicShapeOffPage=Abort|Warn|Ignore
RangeOfPublications=n
...

<OverFlow>
OverFLowFile=Filename
OverFlowSize=Memory

<Custom>
Name=Parameter
Name=Parameter
...
```

(continued) **OPS file format**

Sections, keywords and parameters:

<Generate>

If required, use this section to specify an alternative DOC1 production engine program from that associated with the DOC1GEN program with which OPS was launched.

ProgramLocation — Can be used to indicate an alternative DOC1 Series 5 engine. Where used, *Filename* must indicate a PCOM.DLL file (or equivalent name depending on your production platform) as supplied with versions of DOC1 Generate.

EMFE — Can be used to indicate a DOC1 Series 4 production engine. Where used, *Filename* must indicate an EMFE.EXE file (or equivalent name depending on your production platform). If you use this option you must also code the INI keyword to specify an EMFE initialization file that contains the settings for the job to be launched. If you use the EMFE keyword all other OPS settings are ignored.

ServerMode — Setting the keyword to *True* initiates Server Mode. Setting the keyword to *False* overrides any Server Mode settings contained in the hip file and DOC1GEN runs in batch mode.

<Input>

DataInput — *Filename* specifies the main DOC1 input file; i.e. the keyed record, delimited or XML file that will provide variable data to the current job.

MessageLib — *Filename* specifies the DOC1GEN production message library (PCOMEng.DLL or equivalent). Use this option if the message library file is not in the same location as the DOC1GEN program with which the OPS is to be launched. Also use it if the *ProgramLocation* or *EMFE* keywords are being used to specify an alternative production engine in which case the message file associated with the selected engine should be specified.

MessagesFile — *Filename* specifies the Message1/ Content Author HIM files to be used (note that you can use wildcard characters in the filename, e.g. re*.him). These files contain the required messages.

ResourceHIP — *Filename* specifies the resources file used when publishing output definition separately. Each additional HIP specified relates to a resources HIP targeted at a specific device. These must pair up with entries in the <Output> section.

(continued)	**OPS file format**
ActiveContentLocation	Used when publishing Active Content separately. *Path* specifies the location of the Active Content files. Use %1 in the path as a place holder for the Active Content HIP file name. This name is generated automatically by DOC1 when publishing the Active Content and must not be changed. It is in the form *Rxxxxxxx*, where *x* is an alphanumeric character, for example `R000012A`. If %1 is omitted, the Active Content file name (*Rxxxxxxx*) will be appended to *Path*. This is an acceptable format for Windows and UNIX, but care should be taken on mainframe systems such as z/OS and OS400.
<LicenseInfo>	This section allows you to specify your DOC1 license keycode details for the production platform on which DOC1GEN is to run. If specified this section overrides the keycode options that were specified in the Host object used when publishing a publication (if any).
CustomerName	*String* is the customer name as indicated on your DOC1 keycode report. This is case sensitive and must include any punctuation shown in the report.
Keycode	Code the *Keycode* as shown on your DOC1 keycode report. This must be coded exactly as shown including hyphens. Do not code spaces.
<Journal>	This section allows you to specify or override the file references to be used when creating journal files for the publication. *Name* should be the file alias assigned to a journal object in the publication design. *Filename* should indicate the actual file to receive the journal output. You may code as many entries as necessary to meet the number of journals to be created by the publication.
<LookupTable>	This section allows you to specify or override the file references to be used with lookup table functions within the publication. *Name* should be the file alias assigned to a lookup table function in the publication design. *Filename* should indicate the actual file containing the lookup table data. You may code as many entries as necessary to meet the number of lookup tables referenced by the publication.
<KeyMap>	This section allows you to specify or override the settings used with key maps within the *External Key Maps* section of a production job or Publish Wizard.
Name	The file alias assigned to a key map in the publication design. *Filename* should indicate the actual file containing the key map data. You may code as many entries as necessary to meet the number of key maps referenced by the publication.
Mode	Specifies how the key map is read at run time and is applied to all the key maps associated with a publication. For more details, see " Production job file references" on page 334.

(continued) **OPS file format**

Cached will copy the key map into memory, making access much faster. This is the default setting.
Runtime will access the key map externally each time. This does not use any extra memory, but will not be as fast. You may want to use this option if there are a great number of entries in the map and memory is limited.

<DIJ>

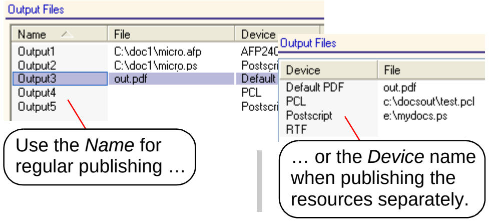
This section allows you to override the settings specified for the DOC1 Interchange Journal in the *Output Files* section of the Publish Wizard. See "Appendix C – DOC1 Interchange Journals" on page 419 for details. *Name* corresponds to the *Name* field which is automatically assigned when using the Publish Wizard. *Filename* corresponds to the *DOC1 Interchange Journal* field and indicates the actual DIJ file you want to use.

<Output>

This section allows you to specify or override the file references to be used when creating the output datastream files produced by the job.

Name

Is the reference name assigned to the output file when publishing the publication. *Filename* indicates the actual file containing the datastream. If set to 'null' the publication will be not be generated for that output device, e.g. output2=null. When specifying output datastream files you may code as many entries as necessary to meet the number of output datastreams to be created by the publication.



Output Files

Name	File	Device
Output1	C:\doc1\micro.afp	AFP240
Output2	C:\doc1\micro.ps	Postscri
Output3	out.pdf	Default
Output4		PCL
Output5		Postscri

Use the *Name* for regular publishing …

Output Files

Device	File
Default PDF	out.pdf
PCL	c:\docsout\test.pcl
Postscript	e:\mydocs.ps
RTF	

… or the *Device* name when publishing the resources separately.

MessageAuditTrail

Filename is the file to be used for Message1 and Content Author audit information for each message that is included in the published documents.

<Trace>

OutputFile

Filename is the name of the file for the trace information. If no file is specified the output is sent to the standard output medium of the Generate host environment (e.g. command prompt window, system log, etc.)

TraceLevel

This controls the amount of trace information that is output.
Off writes nothing i.e. turns trace off.
Default shows the path that Generate took when the error occurred. This is in the form of a tree structure. This enables the error to be tracked down to a particular object in the publication design.
Verbose includes additional information such as internal references and the instructions that are being executed. This can be used by DOC1 Support.

(continued) **OPS file format**

OutputCodePage

This allows you to change the code page for the output file from the *default* host code page to the general Unicode *UTF8* code page.

MemLimit

Controls the maximum amount of memory that the trace module can use for buffering trace information. A value of '0' means no limit.

Publication

Allows the user to specify the *number* of a publication to be traced in addition to the first publication that causes the error. This will only be traced if it occurs before the publication with the error (that causes the trace to stop processing).

<Messages>

MandatoryNotPlaced

Select the action you want DOC1 Generate to take when a mandatory message cannot be included in a document for which it was intended. The default is to *Stop* processing, otherwise you can ignore the error and *Continue* or issue a *Warning* and continue.

MandatoryMessageError

Select the action you want DOC1 Generate to take when a mandatory message has unresolved links – typically when a font used by the message is not included in the resource pack or when a data field used has not been mapped. The default is to *Stop* processing or you can ignore the error and *Continue* processing.

OptionalMessageError

As above but for non-mandatory messages.

CampaignDate

Specifies the date to be used when selecting messages using the activation and expiration attributes as defined within the Message1 and Content Author environments. Options are:
Auto – the current system date
Auto+|-<n><d|w|m> – the current system date plus or minus the specified number of days, weeks or months, e.g Auto+10d, Auto-3w
dd/mm/yyyy or *mm/dd/yyyy* – a specific date

Cycle

String defines which cycle (defined in the Message1 and Content Author environments) to use. If defined, only messages belonging to this cycle will be selected.

MessageProcessing

Allows details for message rejections to be output to the standard output medium for the system on which the program is running. The default is *No*.

NoMessages

Specify the DOC1 Generate action to be taken if messages are expected, but none can be found in the Messages file. The default is to *Stop* processing, or you can ignore the error, issue a *Warning* and continue.

FontMappingFromHip

Setting this to *Yes* will ensure that any mappings that have been applied to fonts used in a publication in the Designer (i.e. are in the HIP file) will also be applied to the same fonts used in messages created in Content Author or Message1.
The default is *No*.

(continued) **OPS file format**

<Server> The keywords in this section are used with the Server Mode method of running DOC1GEN under UNIX and Windows. For more details on Server Mode, see the DOC1 Programmer's Guide.

CommandQueue Including this keyword in the OPS file initiates server mode.
QueueID – is the name of the communication channel in the format applicable to the operating system. Under *UNIX* this is the name of the existing pipe and under *Windows* this is the name of a pipe which will be set up automatically when Server mode is initiated. This must always use the following conventions: `\\.\pipe\name`
address:port – is the name of a communication channel specified as a TCP/IP 'dot' address and an associated port number, such as `10.133.54.202:5000`
hostname:port – is the name of a communication channel specified as a host name known to the current system and an associated number, such as `spa02:5000`

Commandnn – The number *nn* is used to reference the command when the DOC1SBMT program is executed. The *command string* can pass up to nine parameters which can be user defined or predefined elements of the Server Mode environment. The predefined elements are identified by fixed symbols as in the following list (assumes that defaults are being used, i.e. a *ValuePrefix* of "&" and *ValueSuffix* of "."):
&I. – input file name
&Pn. – print file name (where *n* is the file alias assigned to the output file when publishing a publication, i.e. &POutput1. &POutput2.)

&R. – server mode return code…

0 job OK

10 PAGE 1 of N overflow threshold exceeded

15 error executing system command specified by the user

25 failed during processing of job

30 failed during termination of job

50 failed during initialization of processing. Most likely input or print filenames were invalid

&S. – job submission comment generated automatically by Server Mode to identify the process uniquely
&Jn. – journal file (where *n* is the file alias assigned to a journal object, i.e. &JDoc1J1. &JDoc1J2.)
&Dn. – DOC1 interchange journal (where *n* is the file alias assigned to a DIJ object, i.e. &Ddij1.)
&C. – pages generated to the point at which the command is called

CommandBefore Optionally identifies a system command that will be executed prior to the start of processing application data with DOC1GEN. See *Commandnn* for more information.

(continued)	**OPS file format**
CommandOK	Optionally identifies a system command to be executed only after the application data has been processed successfully by DOC1GEN. See *Commandnn* for more information.
CommandFail	Optionally identifies a system command to be executed only when the application data fails to be processed successfully by DOC1GEN. See *Commandnn* for more information.
CommandEnd	Optionally identifies a system command to be executed only after the application data has been processed successfully by DOC1GEN. See *Commandnn* for more information.
AbortOnFail	This defines the behavior of DOC1GEN if a print job run in Server Mode fails. The default *True* will abort Server Mode, while *False* gets Server Mode to attempt to discard the failing print job and await further commands.
<Advanced>	
ErrorFile	This option is used to specify the file that will receive any publication datasets that cannot be processed by DOC1 Generate when a production job is run. It will override the file specified in the *Data record file* option in the Publish Wizard.The *filename* must be in the required format for the operating system.
LogFile	This option is used to specify the file that will receive any error or warning messages issued by DOC1 Generate. The *filename* must be in the required format for the operating system.
Checkpointfile	This option is used to specify the file that will receive the messages that indicate which publication data set is currently being processed.
CPconsole	Checkpointing messages can be reported to the standard output medium of the Generate host environment (e.g. command prompt window, system log, etc.). Set to 0 – do not report messages (default) 1 – switch reporting on.
#Restart	When this is included after Generate has failed, it will restart and continue processing from the last checkpoint using the information in the checkpoint file. See also “Checkpoint progress” on page 342. This can be included anywhere in the OPS file.

(continued)	**OPS file format**
RangeOfPublications	This is used if you want DOC1 Generate to process only a subset of the publication data sets available in the input data file. You may want to do this if you need to rerun portions of a production job without creating a new input data file. You can indicate the sequential numbers of the publication data sets to be processed as follows: `27,280,674` – specific publications `100-1000` – publications between 100 and 1000 `1000+` – all publications after the first 1000 `366,500-1000,2000+` – combinations.
ConstantShapeOffPage	Defines the action DOC1 Generate should take if graphic objects (including text boxes) positioned using constant values for both X and Y offsets are positioned all or in part outside the active logical page area. Options are: *Abort* – Generate aborts immediately. Any output files that have been created by the job are deleted (if this is permitted by the host operating system). *Warn* – a warning message is issued for each object that is found to be positioned outside the logical page area. Processing of the job continues as normal. The off page object is included in the output datastream; the effect of this in the printer/browser environment will depend on the device type. *Ignore* – as above but no warning message is issued. This is the default for objects placed using constant values.
DynamicShapeOffPage	As above but this option applies to graphic objects positioned using variable data for either offset.
<Overflow>	These options are used to override the *Limit composed pages in memory* settings on the *Memory Handling* page in the Publish Wizard. For details, see “Error Handling” on page 343.
OverflowEnabled	When set to *No* – the default value – the *OverflowFile* and *OverflowSize* settings only take effect if the *Limit composed pages in memory* option is specified in the Publish Wizard. When set to *Yes*, the *OverflowFile* and *OverflowSize* settings always take effect.
OverFlowFile	This option designates the temporary file to which the composed files are written.
OverFlowSize	This option specifies the memory limit at which the process of writing to the overflow file begins. The default *memory* value is 4 megabytes. You can use the suffix *K* to indicate kilobytes, *M* to indicate megabytes or if no suffix is used the value will be in bytes.
<Custom>	This section allows you to specify any temporary settings that may be required as part of problem resolution. The keyword *Name* and associated *Parameter* will be provided directly by DOC1 customer support as required. You may code as many entries in this section as necessary.

(continued) **OPS file format**

Examples:

```
<Generate>
ProgramLocation=\\servnet\mt\doc1gen.exe

<Input>
MessageLib=\\servnet\mt\doc1msg.dll
ResourceHIP=C:\Resources\Default pdf.hip
ResourceHIP=C:\Resources\myafp240.hip
ActiveContentLocation=C:\Resources\%1

<LicenseInfo>
CustomerName=Big Corp
Keycode=EN-7824-620B-F1A0-17CC-721F-194D

<Journal>
J1=\trace\doc1j1.txt
J2=\trace\doc1j2.txt
J3=\trace\sn1j3.txt

<LookupTable>
tsub=\\servnet\mt\lookups.txt

<KeyMap>
kmap1=\\resnet\gjk\km1.xml
mode=Runtime

<DIJ>
Output1=\doc1\edm\doc1a.jrn

<Output>
Output1=eas21.afp
Output2=eas21.pdf
Output3=C:\PDF\test1.pdf
Output4=C:\AFP\test1.afp

<Trace>
Outputfile=trace.out
TraceLevel=default
outputcodepage=utf8
memlimit=0
publication=3

<Messages>
MandatoryNotPlaced=Warn
MandatoryMessageError=Continue
OptionalMessageError=Continue
CampaignDate=12/07/2007
Cycle=AC02
MessageProcessing=Yes
NoMessages=Warn
```

(continued)

OPS file format

```
<Server>
CommandQueue=HOST:spa02:5001
Command00=¨sndmsg msg('DOC1GEN processed &I.') tousr(*requester)¨
Command01=¨sndmsg msg('&0.') tousr(*requester)¨
CommandBefore=¨time /t >> start.log¨
CommandOK=¨time /t >> OK.log¨
CommandFail=¨time /t >> Fail.log¨
CommandEnd=¨time /t >> End.log¨
AbortOnFail=False

<Advanced>
ErrorFile=doc1\backups\june21err.txt
LogFile=trace04.out
Checkpointfile=check.out
CPconsole=1
RangeOfPublications=100-350
ConstantShapeOffPage=Warn
DynamicShapeOffPage=Warn

#restart
<OverFlow>
OverFlowFile=doc1\memerror.txt
OverFlowSize=48m

<Custom>
PTF5690=¨Type1¨
```

Using symbols

You can dynamically define parameters used in an OPS file by using symbols when starting a Generate job. The value assigned to a symbol is substituted wherever it is referenced in the OPS file and can be used to provide part or all of any parameter.

Where used, symbols must be defined after any other parameters in the start-up syntax. Under Windows for example, you could specify the following on the command line:

```
doc1gen j1.hip ops=j1.ops ext=txt
```

where *ext* is a symbol name. When referenced in the OPS file the symbol names must be enclosed in percent (%) characters. For example:

```
<Journal>
J1=\trace\doc1j1.%ext%
J1=\trace\doc1j1.%ext%
...
```

Symbol names are case sensitive. Where a symbol is referenced in the OPS but no value is assigned it is treated as an empty string.

NOTE THAT THE FOLLOWING MUST NOT BE USED AS SYMBOL NAMES: OPS, MODE, ECP, MMGX.

Symbols may also be defined within the OPS file itself by coding them in an <OPS> section prior to where they need to be referenced. The format is as follows:

```
<Symbols>
Name =Parameter
Name =Parameter
...
```

If the same symbol name is specified both in start-up parameters and in the OPS itself then the start-up parameter will override. For example; in the OPS:

```
<Symbols>
RunName=run
BaseDir=\doc1

<Journal>
J1=%BaseDir%\trace\doc1j1.txt
J2=%BaseDir%\trace\doc1j2.txt
J3=%BaseDir%\trace\%RunName%j3.txt

<Output>
Output1=%RunName%1.afp
Output2=%RunName%1.pdf
```

and on the Windows command line:

```
doc1gen in.hip OPS=OFile RunName=tst
```

the ‘J3’ journal name will be specified as `\doc1\trace\tst.txt`.

DOC1GEN under z/OS

Preparation: A DOC1GEN job is typically submitted to the system via standard JCL.

The JOBLIB and/or STEPLIB concatenation must reference the main DOC1 load library and message library datasets. You may also need to include references to the IBM Language Environment (LE) run-time libraries if these are not known to system libraries.

The HIP file that controls the job plus an OPS file (where used) are specified as parameters on the EXEC card. However, these normally indicate DD references that are resolved to dataset names in subsequent DD cards.

The JCL must also include DD cards for all other files that have been identified using DD references in the HIP or OPS.

No two output files should be members of the same dataset.

EXEC card syntax:

```
EXEC PGM=DOC1GEN,PARM=('DD:HipRef [,OPS=DD:OpsRef ] [,ECP=DD:EcpRef ] [,SCP=CodePage ]
[,#restart] [,symbols]')
```

Parameters:

HipRef is the DD label indicating the DOC1 HIP file that will control the job.

OpsRef is the DD label indicating an override production settings file if appropriate.

EcpRef is the DD label indicating the DOC1 Extended Code Page file which will be required for most non-Western applications. The ecp file must be placed in a fixed block dataset.

CodePage is the number of a host code page to be used instead of the default code page – US (37).

#restart restart the job from the last checkpoint position. See “Checkpoint progress” on page 342.

symbols See “Using symbols” on page 361.

Example JCL

```
//DOC1JOB6 JOB  '5438','JDOE',CLASS=F,REGION=2M
//DOC1GEN  EXEC PGM=DOC1GEN,
//      PARM='DD:DOC1HIP,OPS=DD:DOC1OPS,ECP=DD:DOC1ECP'
//*Generate load libraries. You may need to add run-time libs
//STEPLIB  DD DISP=SHR,DSN=PROD.DOC1.LOAD
//         DD DISP=SHR,DSN=PROD.DOC1.MSGS
//*HIP & OPS files
//DOC1HIP  DD DISP=SHR,DSN=PROD.DOC1.RUN(JOB6HIP)
//DOC1OPS  DD DISP=SHR,DSN=PROD.DOC1.RUN(NEWFILES)
//*Extended Code Page file
//DOC1ECP  DD DISP=SHR,DSN=PROD.DOC1.RUN(DOC1ECP)
```

```
//*Input data (as per DD ref in HIP or OPS)
//DOC1INPT DD DISP=SHR,DSN=PROD.DOC1.DATA
//*Lookup tables (as per DD refs in HIP or OPS)
//DOC1TL1  DD DISP=SHR,DSN=PROD.DOC1.RUN(JOB6TL1)
//DOC1TL2  DD DISP=SHR,DSN=PROD.DOC1.RUN(JOB6TL2)
//*Output datastreams (as per DD refs in HIP or OPS)
//AFPOUT1  DD SYSOUT=X,DCB=LRECL=8205
//AFPOUT2  DD SYSOUT=X,DCB=LRECL=8205
//*Journals (as per DD refs in HIP or OPS)
//DOC1JRN1 DD DISP=SHR,DSN=PROD.DOC1.RUN(JOB6JRN1)
//DOC1JRN2 DD DISP=SHR,DSN=PROD.DOC1.RUN(JOB6JRN2)
```

DOC1GEN under UNIX, Windows and OpenVMS

Preparation: DOC1GEN is executed from the command prompt.

The HIP file that controls the job plus an OPS file (where used) are identified as parameters to the start up command.

All other references to files to be used or created by Generate are defined within the HIP or OPS files. You will need to ensure that these are available (or creatable) at the locations indicated.

Syntax:

```
doc1gen HipRef [ops=OpsRef] [ecp=EcpRef] ["#restart"] [symbols]
```

Parameters:

HipRef is the path/file name of the DOC1 HIP file that will control the job.

OpsRef if an override production settings file is being used this is the path/file name of the OPS.

EcpRef is the path/file name of the DOC1 Extended Code Page file which will be required for most non-Western applications.

#restart restart the job from the last checkpoint position. See “Checkpoint progress” on page 342. Note that the quotes are only required when running under UNIX.

symbols See “Using symbols” on page 361.

Examples

For Windows:

```
C:\doc1\run\doc1gen job6.hip ops=C:\doc1temp\newfiles.ops ecp=doc1ecp
```

For UNIX:

```
/doc1/run/doc1gen job6.hip ops=C/doc1temp/newfiles.ops ecp=doc1ecp
```

For OpenVMS (assuming that the program name has been defined as a symbol):

```
doc1gen DISK1:[doc1gen]job6.hip ops=[doc1gen]newfiles.ops
```

DOC1GEN under OS/400

Preparation: DOC1GEN is executed using a direct program call.

Syntax:

```
CALL DOC1GEN PARM ('HipRef' ['OPS=OpsRef'] ['ECP=EcpRef'] [,SCP=CodePage ] [,#restart]
[symbols])
```

Parameters:

HipRef is the library/file/member name of the DOC1 HIP file that will control the job.

OpsRef if an override production settings file is being used this is the library/file/member name of the OPS.

EcpRef is the library/file/member name of the DOC1 Extended Code Page file which will be required for most non-Western applications.

CodePage is the number of a host code page to be used instead of the default code page – US (37).

#restart restart the job from the last checkpoint position. See “Checkpoint progress” on page 342.

symbols See “Using symbols” on page 361.

Trouble Shooting In certain circumstances where DOC1GEN has earlier finished abnormally some files may remain locked by the operating system. This situation may be evidenced by messages such as ″File… was already closed…″ and ″Member… already in use…″. If this occurs you can use the WRKJOB utility to display locks and then clear each locked object using DLCOBJ. Refer to your OS/400 documentation for more information.

Example

```
CALL DOC1GEN PARM ('JOB6HIP' 'OPS=OPSLIB/EARLY(NEWFILES)' 'ECP=DOC1/INST(DOC1ECP)')
```

Administrative and Repository Tasks

Access control

A user who creates a new DOC1 design repository (as part of installing the DOC1 Work Center software) is automatically added to the Administrators group and is given full access to all objects within the repository.

Other users that require access that repository must be registered as a user by an Administrator. Users can be grouped to make access right assignments more convenient.

By default, design objects are available to all users. You can create projects folders in which design objects are stored and specify which users have access to such folders and the level of access they should have.

User accounts

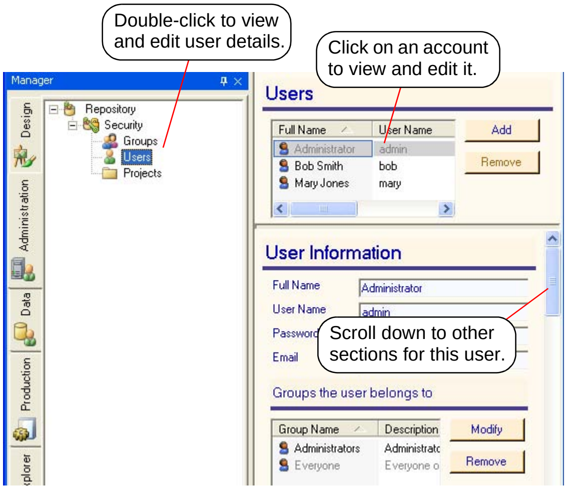
Before any user other than the initial Administrator can log in to the repository they must be assigned a user account.

Each account can have one or more **Projects** associated with it. This allows access to design resources within the named projects.

Additionally you can confer project rights by including user accounts within **Groups**. The project rights defined for the group are conferred on each user belonging to it.

Only an administrator has full control over creating and deleting user accounts, as well as changing someone's *user name*. Other users are restricted to editing their own *full name, e-mail* and *password.*

▶ **To work with user accounts:** in the Work Center Manager select the Administration tab and expand the tree in the navigator to show *Users* within the *Security* group. In the *Users* list click on an account to view and edit its details in the workspace.



Double-click to view and edit user details.

Click on an account to view and edit it.

Scroll down to other sections for this user.

Users

Full Name	User Name
Administrator	admin
Bob Smith	bob
Mary Jones	mary

User Information

Full Name: Administrator

User Name: admin

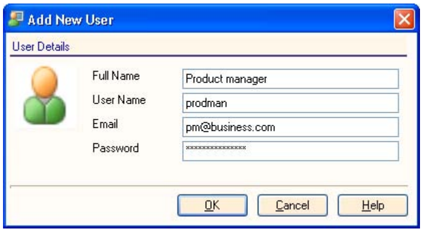
Password

Email

Groups the user belongs to

Group Name	Description
Administrators	Administratc
Everyone	Everyone o

▶ **To create a new user account:** access the user account details and in the Users list click *Add*. In the *Add New User* dialog enter the basic information about the new account and click *OK*. The account is created with default attributes.

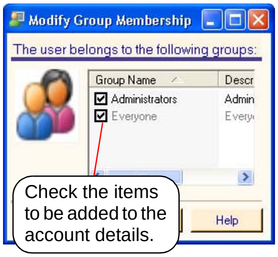


Add New User

User Details

Full Name	Product manager
User Name	prodman
Email	pm@business.com
Password	***************

▶ **To associate a user with groups or projects:** access the relevant user account details and click *Modify* in the appropriate section, scrolling down where necessary. A dialog will show existing items and allow you to select those to be added to the account details.



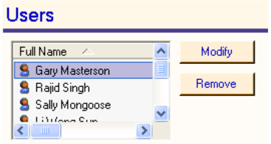
Modify Group Membership

The user belongs to the following groups:

Group Name	Descr
☑ Administrators	Admin
☑ Everyone	Everyo

Check the items to be added to the account details.

▶ **To delete a user account:** highlight the accounts you want to delete in the *Users* list and click *Remove*.



Users

Full Name
Gary Masterson
Rajid Singh
Sally Mongoose
Li Wong Sun

User groups

User groups provide the same access control functions as individual user accounts but apply to all accounts assigned to a group. If user accounts have individual rights these will apply in addition to the group rights.

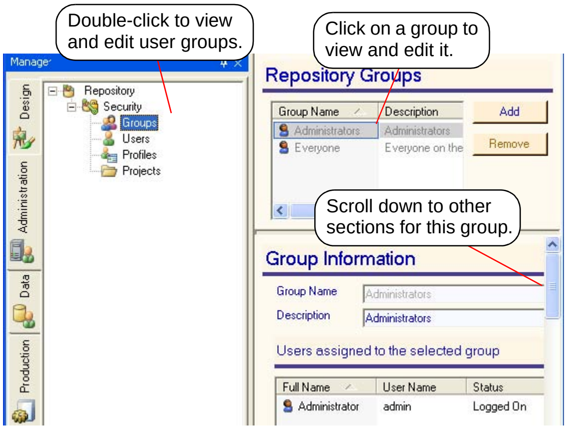
There are two built-in groups:

Administrators – users assigned to this group can perform repository administration tasks. A user who creates a new DOC1 design repository is automatically added to the Administrators group.

Everyone – by default, all users are added to this group.

You can create new groups as required but these built-in groups should not be deleted. You can add as many users as required to a group but you cannot add one group to another (i.e. create nested groups).

▶ **To work with user groups:** in the Work Center Manager select the Administration tab and expand the tree in the navigator to show *Groups* within the *Security* group. In the *Repository Groups* list click on a group to view and edit its details in the workspace.



Double-click to view and edit user groups.

Click on a group to view and edit it.

Scroll down to other sections for this group.

Repository Groups

Group Name	Description
Administrators	Administrators
Everyone	Everyone on the

Group Information

Group Name: Administrators

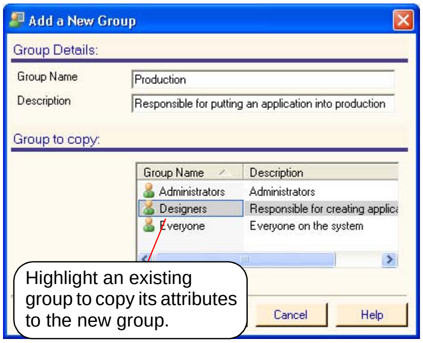
Description: Administrators

Users assigned to the selected group

Full Name	User Name	Status
Administrator	admin	Logged On

▶ **To create a new user group:** access the user groups and in the Repository Groups list click *Add*. In the *Add New Group* dialog box enter a name and descriptive comment for the group and click *OK*. The group is created with default attributes.

▶ **To edit group membership and attributes:** access the user groups and click on the group to be edited. Edit as required in the Group Information section.



Add a New Group

Group Details:

Group Name: Production

Description: Responsible for putting an application into production

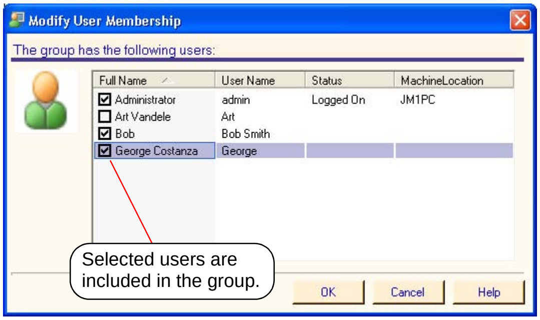
Group to copy:

Group Name	Description
Administrators	Administrators
Designers	Responsible for creating applica
Everyone	Everyone on the system

Highlight an existing group to copy its attributes to the new group.

▶ **To update the user list:** select the group to be updated and click *Modify*.

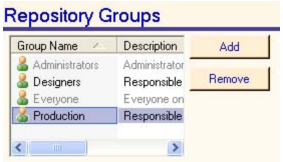
▶ **To associate the group with projects:** click *Modify* in the Projects section and a dialog box listing available projects is displayed. Select the required project and click *Open*.



Full Name	User Name	Status	MachineLocation
☑ Administrator	admin	Logged On	JM1PC
☐ Art Vandele	Art		
☑ Bob	Bob Smith		
☑ George Costanza	George		

Selected users are included in the group.

▶ **To delete a user group:** in the *Groups* list highlight the group you want to delete and click *Remove*.



Group Name	Description
Administrators	Administrator
Designers	Responsible
Everyone	Everyone on
Production	Responsible

Projects

Projects are containers within which publication design objects can be created and maintained. A project can contain one or more sub-projects enabling more detailed levels of object organization if required.

A project provides an access control mechanism for the design resources it contains. When a project is created each user in the system has universal access to it, providing them with complete access rights to resources contained within it. Adding a user or group to a project restricts access to those associated with it. Each user or group of users has a specific set of access rights from the following list:

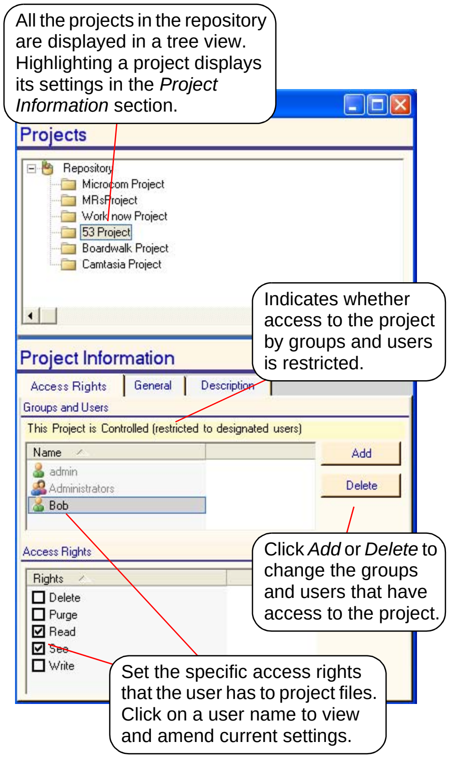
See – the user can see project objects in the Work Center Manager

Read – the user can browse the content of project objects

Write – the user can edit project objects. (Note that when users come to save new design objects they will only be presented with a list of project folders that they are permitted to work with.)

Delete – the user can send project objects to the recycle bin.

Purge – the user can delete project objects from the recycle bin.



This Project is Controlled (restricted to designated users)

Name
admin
Administrators
Bob

☐ Delete
☐ Purge
☑ Read
☑ See
☐ Write

All the projects in the repository are displayed in a tree view. Highlighting a project displays its settings in the *Project Information* section.

Indicates whether access to the project by groups and users is restricted.

Click *Add* or *Delete* to change the groups and users that have access to the project.

Set the specific access rights that the user has to project files. Click on a user name to view and amend current settings.

▶ **To create a project:**

1. In the Work Center Manager select any tab that relates to design objects. *Design*, *Production* or *Data*.
2. If you want to create a sub-project first open the master project in which it is to be created.
3. From the File menu select *New Project*.

Name
LoanStatements
CurrentAccounts

▶ **To control access to a project:** explicitly add a user or group to a project via the *Access Rights* tab in the *Project Information* section.

▶ **To move objects into or between projects:** in the Project Explorer select the object you want to move and select *Edit/Copy* or click 📋. Browse to the project you want to move the object into and select *Edit/Paste* or click 📋.

Managing Resources

Resources stored in the DOC1 design repository are considered to be one of two types: composite and non-composite. Composite resources are objects such as publications and documents that make reference to other resources stored in the repository.

The DOC1 Work Center provides a range of features for managing and maintaining the resources within the repository. As resources are *issued* new object revisions are automatically created. You are able to restore older revisions of individual objects and roll-back composite resources to restore an group of objects to a particular revision level.

You can copy resources between projects which creates a new version of the resources to be managed separately. You can rename and delete resources from project folders and you can manually lock a resource file for an indefinite period.

Private/public resources

Before an object is *issued* it is always considered to be private. This means that it cannot be used within a composite object. Most resources automatically become public as soon as they are *issued* and can then be used in composite objects.

ⓘ This should not be confused with allowing access rights for a particular resource to other users. Access rights can be conferred in project settings regardless of the public/private state.

The exceptions to this rule are document designs and images that have been included directly into a publication and not imported into the repository. Both are always considered to be private to an individual publication unless they are specifically made public in the repository.

▶ **To make a document public:** from the Document Editor select *View/Publication* or right-click in the logic map and click *View Publication*. Highlight the document and select *Edit/Document/Make Public*.

ⓘ Once a document has been made public it can no longer be edited from within a publication. To edit a public document, you must open it from the Document bar in the Design tab.

▶ **To make an image public:** open the document, select the image object and in the *Edit* menu select *Make public*.

Working with resource objects

Objects can be managed in the DOC1 Work Center Manager window.

▶ **To open the Work Center manager:** from the *View* menu select *Manager* and click *All*.

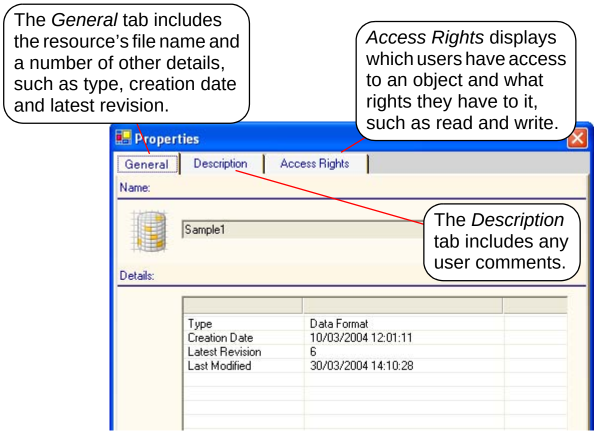
FOR MORE INFORMATION SEE "USING THE WORK CENTER MANAGER" ON PAGE 21.

Individual resources can be selected for subsequent uses by highlighting them in the tree view or the details window. Standard windows selection techniques can be used to select multiple resources. From the *Edit* menu you can choose *Select All* to highlight all the files in the currently selected window.

Resource properties

Each object in the repository has properties that provide detailed information about the resource.

▶ **To view resource properties:** highlight the resource that you want to view and select *File/Properties*.
–or–
right-click the resource and select *Properties* from the shortcut menu. The *Properties* dialog box is displayed. Select the tab corresponding to the information that you want to view.



Name: Sample1

Details:

Type	Data Format
Creation Date	10/03/2004 12:01:11
Latest Revision	6
Last Modified	30/03/2004 14:10:28

The *General* tab includes the resource's file name and a number of other details, such as type, creation date and latest revision.

Access Rights displays which users have access to an object and what rights they have to it, such as read and write.

The *Description* tab includes any user comments.

Copying

Copying resources creates new versions of the selected objects that are independent of the originals. Objects are copied from their original folders and then pasted into new folders. Resources must be public before they can be copied between folders in the project explorer.

▶ **To copy a resource to a new location:** from the navigator highlight the resources you want to copy and select *Copy* from the *Edit* menu. Move to the folder where you want to copy the objects and select *Paste* from the *Edit* menu.

Deleting and restoring

Deleting a resource in the Work Center will remove it from the project it was in, but does not remove it from the design repository. The deleted resource will be placed in the recycle bin. Resources remain in the recycle bin until they are purged or restored. Any object in the recycle bin can be restored to its previous location in the repository but once resources have been purged they cannot be restored. To purge or restore resources from the recycle bin you must be logged on as the administrator or have the *Perform Backup and Restore* operations privilege. Note that you can only delete resources that are not used by anything else.

▶ **To delete a resource:** highlight the file to be deleted in the navigator. From the *File* menu select *Delete*. The resource will be removed to the recycle bin in the repository.
– or –
right-click the file to be deleted in the navigator and select *Delete* from the shortcut menu to remove the file to the recycle bin in the repository.

▶ **To purge resources from the Work Center:** from the project explorer double-click the recycle bin. Highlight the resources and select *File/Purge*. To purge all the resources from the recycle bin, highlight it and select *File/Empty Recycle Bin*.

▶ **To restore a resource from the recycle bin:** from the project explorer double-click the recycle bin. Highlight the resource or resources you want to restore from the recycle bin and select *Restore* from the *File* menu.

Renaming

Changing resource names does not affect the relationship between files as DOC1 keeps track of file associations through unique identifiers imbedded in each file.

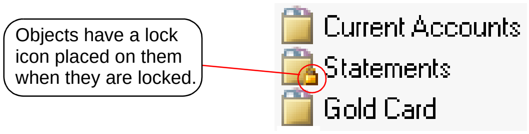
▶ **To rename a resource:** right-click the resource in the Navigator and select *Rename* from the shortcut menu. Edit the name and press Enter.
– or –
highlight the resource in the navigator and select *Rename* from the *File* menu. Edit the name and press Enter.

Locking

Resources that are opened for editing are automatically locked. By default, when the object is closed the lock is removed. If required, DOC1 allows you to keep a resource locked from access by other users even when it is not open for edit. If you try to open a locked resource in a client/server environment a message box will display which user has locked the item.

The Work Center administrator has the ability to lock or unlock resources while other users in the system are editing them.

▶ **To lock/unlock a resource:**
highlight the resource in the navigator. From the *Edit* menu select *Lock* or *Unlock* as needed.


Objects have a lock icon placed on them when they are locked.

Viewing and editing

Resources can be opened for viewing or editing by selecting them from either the navigator or the details view in the Work Center. The *Open Read-only* and *Open* options available from the *File* menu will open the resource in the appropriate editor. Selecting the *Open* option will enable you to make changes to the resource, while *Open Read-only* provides a read–only copy of the resource showing the last issued version. As well as viewing the actual resource the properties, history and file associations of resources can be viewed.

▶ **To view a resource:** select *File/Open Read-only*. Browse to the file you want to view and click *Open*. The file will be opened in its appropriate editor.

> If the file type is unknown the Windows *Open With* dialog box will be displayed, prompting you for a program with which to open the file.

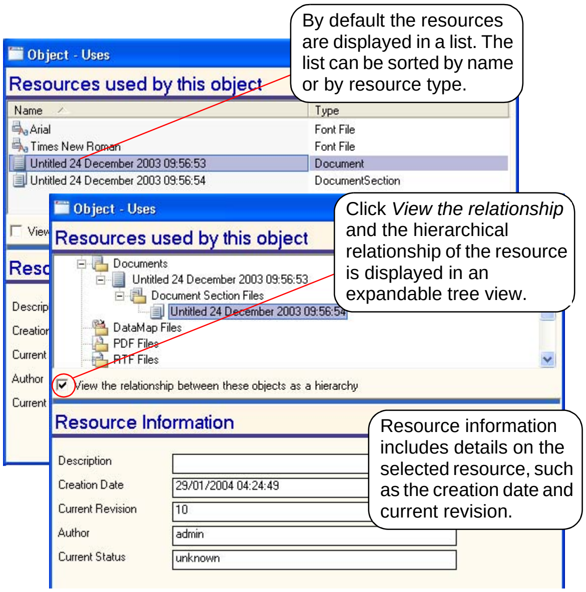
▶ **To edit a resource:** double-click on the resource you want to edit in the navigator or the details view/workspace and it will be opened in its appropriate editor.

> For more information on Editing and Creating Resources, see "About Design Files and Resources" on page 83.

Displaying object associations

The Work Center provides the ability to view the associations a resource has with other objects in the repository plus a history of such associations.

▶ **To show the resources used by a resource:** select the resource from the navigator and select *Show Uses* from the *Tasks* menu.



Object - Uses

Resources used by this object

Name	Type
Arial	Font File
Times New Roman	Font File
Untitled 24 December 2003 09:56:53	Document
Untitled 24 December 2003 09:56:54	DocumentSection

Object - Uses

Resources used by this object

- Documents
  - Untitled 24 December 2003 09:56:53
    - Document Section Files
      - Untitled 24 December 2003 09:56:54
- DataMap Files
- PDF Files
- RTF Files

☑ View the relationship between these objects as a hierarchy

Resource Information

Description	
Creation Date	29/01/2004 04:24:49
Current Revision	10
Author	admin
Current Status	unknown

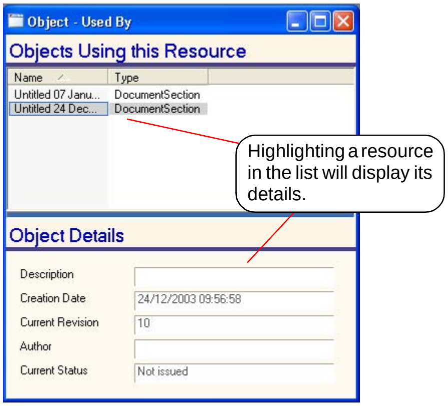
By default the resources are displayed in a list. The list can be sorted by name or by resource type.

Click *View the relationship* and the hierarchical relationship of the resource is displayed in an expandable tree view.

Resource information includes details on the selected resource, such as the creation date and current revision.

▶ **To show the objects using a resource:** select the resource from the navigator and select *Show Used By* from the *Tasks* menu. The Object–Used By details are displayed in the workspace. It contains a list of resources sorted by name or type. Highlighting a resource displays its details.

▶ **To show the history of a resource:** select the resource from the navigator and select *Show History* from the *Tasks* menu. The *Show History* dialog box is displayed. Select the required settings and click *Finish*. The Object–History details are displayed in the workspace.


Object - Used By

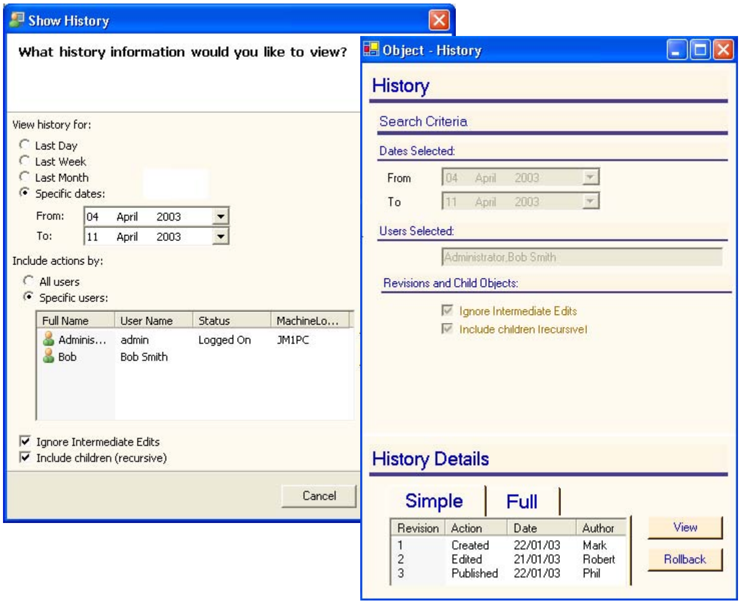
Objects Using this Resource

Name	Type
Untitled 07 Janu...	DocumentSection
Untitled 24 Dec...	DocumentSection

Highlighting a resource in the list will display its details.

Object Details

Description	
Creation Date	24/12/2003 09:56:58
Current Revision	10
Author	
Current Status	Not issued


Show History

What history information would you like to view?

View history for:

○ Last Day
○ Last Week
○ Last Month
● Specific dates:

From: 04 April 2003
To: 11 April 2003

Include actions by:

○ All users
● Specific users:

Full Name	User Name	Status	MachineLo...
Adminis...	admin	Logged On	JM1PC
Bob	Bob Smith		

☑ Ignore Intermediate Edits
☑ Include children (recursive)

Object - History

History

Search Criteria

Dates Selected:

From 04 April 2003
To 11 April 2003

Users Selected:

Administrator,Bob Smith

Revisions and Child Objects:

☑ Ignore Intermediate Edits
☑ Include children (recursive)

History Details

Simple | Full

Revision	Action	Date	Author
1	Created	22/01/03	Mark
2	Edited	21/01/03	Robert
3	Published	22/01/03	Phil

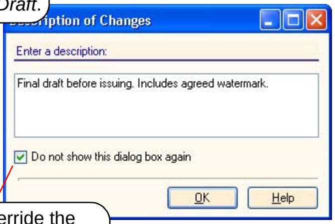
Revision control

Until a design resource is *issued* it is considered to be a *draft* version. Any saved changes to a draft version will overwrite the previous draft. Once a resource is issued a new revision of the object is stored in the repository. Further changes can then be made to a new draft version before reissuing and so on.

ONLY DESIGN OBJECTS ARE REVISION CONTROLLED. ADMINISTRATION OBJECTS SUCH AS USERS AND USER GROUPS HAVE AN AUDIT TRAIL BUT ARE NOT ISSUED AND DO NOT MAINTAIN REVISIONS.

Draft versions can be made of publications, documents and Active Content objects. When you save a draft you have the option of including a description for it. This information can be used to ensure that you revert to the correct draft version. For more details, see “Work Center Preferences” on page 24.

This dialog box is invoked when you choose to *Save Draft*.



Enter a description:

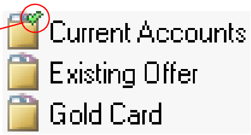
Final draft before issuing. Includes agreed watermark.

☑ Do not show this dialog box again

Select this to override the Work Center Preferences option to *Prompt for description to draft changes* for the resource being saved.

Other resources that are not editable within the DOC1 Work Center can be issued after they have been imported. However, editable resources like publications, documents and Active Content can be issued at the same time as saving them.

▶ **To issue an object:** select it in the Work Center Manager and select *Issue* from the *Tasks* menu. The object and all its referenced files are issued.

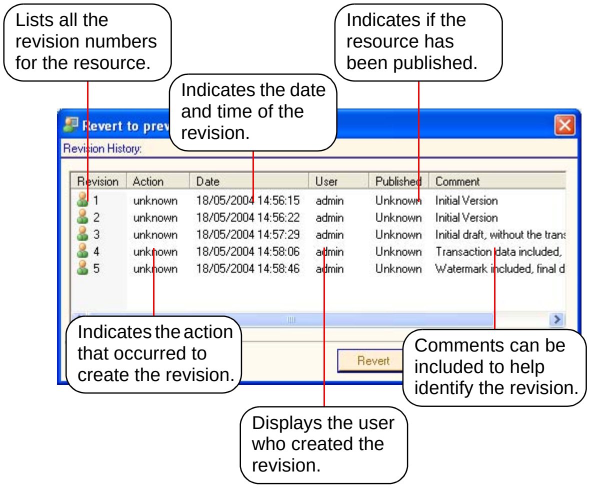


Objects have a check mark placed on them when they have been issued.

Revision history

A revision history lists all of an object's revisions and represents the occasions on which a resource has been issued.

▶ **To view a resource's revision history:** select an object file that you want to view and select *Revert* from the *Edit* menu. The *Revisions* dialog box is displayed showing all the revisions for the corresponding resource.



Lists all the revision numbers for the resource.

Indicates the date and time of the revision.

Indicates if the resource has been published.

Revision	Action	Date	User	Published	Comment
1	unknown	18/05/2004 14:56:15	admin	Unknown	Initial Version
2	unknown	18/05/2004 14:56:22	admin	Unknown	Initial Version
3	unknown	18/05/2004 14:57:29	admin	Unknown	Initial draft, without the trans
4	unknown	18/05/2004 14:58:06	admin	Unknown	Transaction data included,
5	unknown	18/05/2004 14:58:46	admin	Unknown	Watermark included, final d

Indicates the action that occurred to create the revision.

Comments can be included to help identify the revision.

Displays the user who created the revision.

Restoring revisions

You can restore any object back to the state it was in after any issue. For composite objects you will need to use the **Rollback** option to ensure that the other objects it contains are also restored to the correct revision level. For other objects the **Revert** function provides a simple method of restoring a previous revision.

Restoring a previous revision of an object automatically creates a new 'latest' revision. No revisions of a resource are deleted by this process.

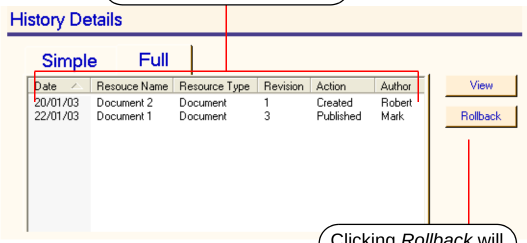
▶ **To revert to a previous revision:**

1. Select the object to be reverted in the Work Center Manager.
2. Select *Revert* from the *Edit* menu and the *Revert to Previous Revision* dialog box is displayed.
3. Select the required revision and click *Revert*.

▶ To rollback to a previous revision:

1. Select the object to be rolled-back in the Work Center Manager and from the *Tasks* menu select *Show History*. The *Show History* dialog box is displayed.
2. Set the history search options as required and click *Finish* to load the History view in the main workspace.
3. In the *History Details* pane select the revision you want to rollback to and click *Rollback*.

History details are displayed in *Simple* and *Full* form and list the necessary information to select the correct version for rollback. Full form is shown here.



Date	Resouce Name	Resource Type	Revision	Action	Author
20/01/03	Document 2	Document	1	Created	Robert
22/01/03	Document 1	Document	3	Published	Mark

Clicking *Rollback* will make the highlighted revision the new current revision.

Repository management

The DOC1 Design Repository stores and controls revisions of all the design resources and other objects that are used in the DOC1 Work Center. These include resources that are created within the Work Center itself or have been imported into it from an external source.

The repository consists of a *database* and a *filestore*. The *database* is used to provide revision control for the repository and will be in the default location as specified in the SQL Server. The *filestore* is where DOC1 design files and resources are stored, and is specified as a folder location.

DOC1 Repository Configuration

Administrative tasks can be performed on the repository using the DOC1 Repository Configuration Tool. For instance you can install a new repository, upgrade, backup and restore existing repositories, and enable different repositories to be used.

The tool is supplied with DOC1 distribution material and is run separately.

▶ **To run the Repository Configuration Tool:**

1. Ensure that there are no users using the repository.
2. From the *Start* menu select *Programs\PBBI CCM\DOC1 Repository Configuration Tool* (or *Programs\Group1 …)*
3. In the *Connect to SQL Server* dialog box click the *Browse* button to select the server that is being used for the repository from the *SQL Server* list – or *local* if it is on your machine. If you are connecting to an SQL server that requires its own login, i.e. not the Windows user name and password, clear the *Use Windows authentication* check box and enter the SQL *user name* and *password*.
   Click *Connect*.
4. Select the required task from the tabs along the top of the window. More detailed instructions for each task are given below.

Setting up the repository

A repository is configured and set up as part of the DOC1 installation process. The Repository Configuration Tool also allows you to install a new repository, and upgrade or change the settings of an existing repository.

You can install any number of repositories, selecting which one to use when logging in to DOC1 – see “Repository selection configuration” on page 383.

▶ To install a new repository:

1. In the Repository Setup page, click *Install a new repository*.
2. Enter the *Database details*.

   If you specify an existing database name, you will be given the option to overwrite the database.
   The database itself will be in the location that was specified in the SQL Server. You can specify another location but it is not recommended.

   Note that the backup locations for the database and the filestore used with the backup/restore options must be accessible by the database server. For Client/Server environments the accessible location is determined by the SQL Server account. The default locations are based on the DOC1 installation folder.
   Note that any existing backup files will not be overwritten.
3. Enter the *Filestore details*.

   Use a different filestore location for each repository.
4. Click *Create*. If a list of repositories has been set up (see below) you will be given the option of adding this new repository to it.
   The log window will show the various stages of the installation.

▶ To upgrade an existing repository:

1. In the Repository Setup page, click *Upgrade an existing repository*.
2. Backup or compact the database if required.

   Setting the option to *compact the database* will defragment the database and rebuild its index. This will reduce the size of the database, but can take some time, especially for large repositories.

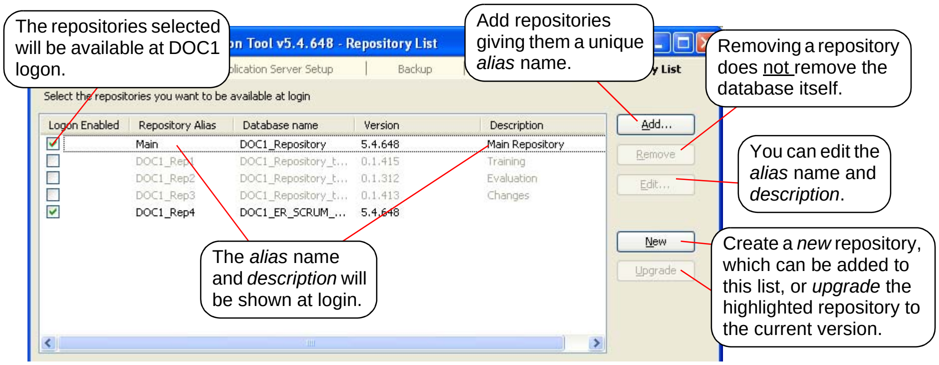
   ⓘ IF YOU HAVE LARGE AMOUNTS OF DATA USE THE COMPACT OPTION WITH CAUTION AS COMPACTING THE DATABASE MAY SLOW DOWN PERFORMANCE WHEN USING DOC1.
3. Select a database to be upgraded from the drop down list. Use the browse buttons to change the backup folder locations.
4. Click *Upgrade*. The log window will show the various stages of the installation.

Repository selection configuration

DOC1 can be run with different repositories – for instance, you could have separate ones for production, training and evaluation. The repositories must all be on the same SQL server, that is accessible from DOC1. You can select which repository to use when logging on to DOC1.

▶ **To create a list of repositories to be available at login:** select the *Repository list* tab. Add any relevant repositories to the list, and select only those that are to be available at login.

> ⓘ REPOSITORIES APPEAR SHADED WHEN THEY ARE OUT OF DATE. IF YOU WANT TO SELECT ONE OF THESE YOU MUST FIRST UPGRADE TO THE LATEST VERSION. SEE PAGE 382 FOR DETAILS.



Select the repositories you want to be available at login

Logon Enabled	Repository Alias	Database name	Version	Description
☑	Main	DOC1_Repository	5.4.648	Main Repository
□	DOC1_Rep1	DOC1_Repository_t...	0.1.415	Training
□	DOC1_Rep2	DOC1_Repository_t...	0.1.312	Evaluation
□	DOC1_Rep3	DOC1_Repository_t...	0.1.413	Changes
☑	DOC1_Rep4	DOC1_ER_SCRUM_...	5.4.648	

The repositories selected will be available at DOC1 logon.

Add repositories giving them a unique *alias* name.

Removing a repository does not remove the database itself.

You can edit the *alias* name and *description*.

The *alias* name and *description* will be shown at login.

Create a *new* repository, which can be added to this list, or *upgrade* the highlighted repository to the current version.

Running repository backup

Backing up the repository will take a snapshot of the entire repository. This can provide protection in case of data loss or corruption.

> ⓘ YOU MUST ENSURE THAT NO USERS ARE LOGGED ONTO THE REPOSITORY WHEN THE BACKUP IS RUN PARTICULARLY IN A CLIENT/SERVER ENVIRONMENT.

The backup process can create the following files:

.dbk – this file contains the contents of the database

.zip – this file contains the contents of the filestore and when the archive option is selected it contains both the database and filestore backups

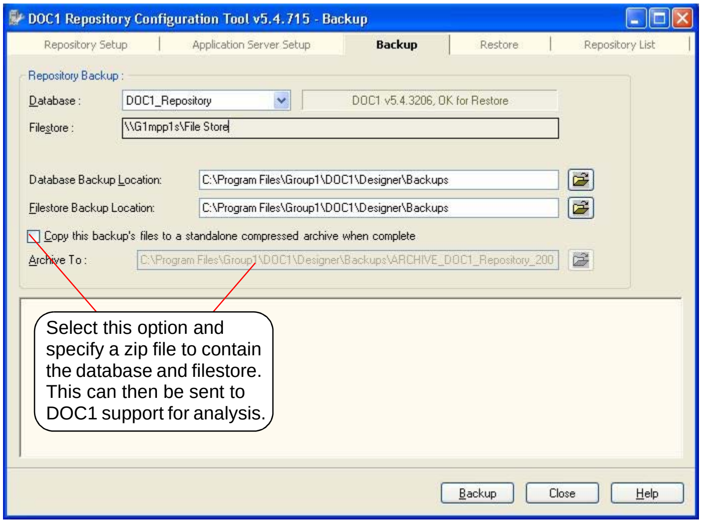
The filenames for the backup files are controlled by DOC1 and cannot be set by the user. A unique sequential number is added to the name to differentiate between backups.

The filenames use the following structure:
RepositoryName_Backup#.%1

For example:
DOC1Repository_3.zip

▶ To backup the repository:

1. Ensure that the Repository Configuration Tool is connected to the relevant server.
2. Select the *Backup* tab and from the *Database* list select the database that you want to backup.
3. Click the browse button 📂 to select a location where the database and filestore contents will be placed for backup.
4. To create a separate zip file containing the database and filestore contents select *Copy Backup Files to Archive* and then click the browse button 📂 to select a location for the separate compressed archive of the database.
5. Click *Backup* and the designated backup files are created.



DOC1 Repository Configuration Tool v5.4.715 - Backup

Repository Backup :

Database : DOC1_Repository   DOC1 v5.4.3206, OK for Restore

Filestore : \\G1mpp1s\File Store

Database Backup Location: C:\Program Files\Group1\DOC1\Designer\Backups

Filestore Backup Location: C:\Program Files\Group1\DOC1\Designer\Backups

☐ Copy this backup's files to a standalone compressed archive when complete

Archive To : C:\Program Files\Group1\DOC1\Designer\Backups\ARCHIVE_DOC1_Repository_200

Select this option and specify a zip file to contain the database and filestore. This can then be sent to DOC1 support for analysis.

Restoring the repository

You must ensure that no users are logged onto the design repository when the database is being restored – particularly in a Client/Server environment. Some of the restore options will only be available if you use SQL Server for your database.

▶ To restore the repository:

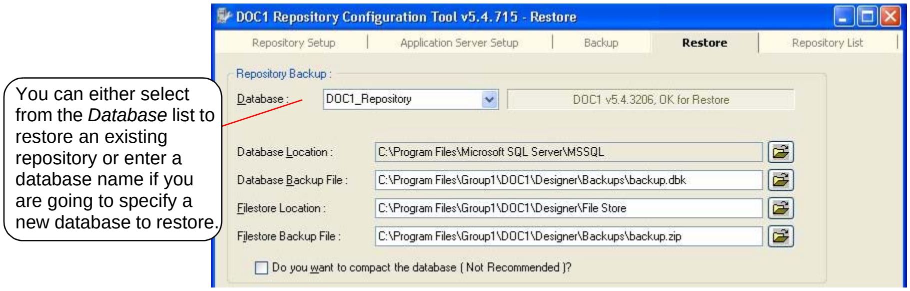
1. Ensure that the Repository Configuration Tool is connected to the relevant server.
2. Select the *Restore* tab and from the *Database* list select the database that you want to restore
   – or –
   Enter the name of a new database that you want to restore.

ⓘ YOU CANNOT USE A UNIVERSAL NAMING CONVENTION (UNC) FILEPATH WITH THE BACKUP FILES WHEN RESTORING A REPOSITORY.

3. Select the required files for the restore. Click the browse button 📂 to select the files for the following:

   - *Database Location* – the location where you want to restore the database
   - *Database Backup File* – the database (.dbk) file that you want to restore
   - *Filestore Location* – the location where you want to restore the filestore
   - *Filestore Backup File* – the filestore (.zip) file that you want to restore

4. Compact the database if required. Use with caution – see page 382 for details.
5. Click *Restore* and the database and filestore will be restored.



You can either select from the *Database* list to restore an existing repository or enter a database name if you are going to specify a new database to restore.

DOC1 Repository Configuration Tool v5.4.715 - Restore

Repository Backup :

Database : DOC1_Repository — DOC1 v5.4.3206, OK for Restore

Database Location : C:\Program Files\Microsoft SQL Server\MSSQL

Database Backup File : C:\Program Files\Group1\DOC1\Designer\Backups\backup.dbk

Filestore Location : C:\Program Files\Group1\DOC1\Designer\File Store

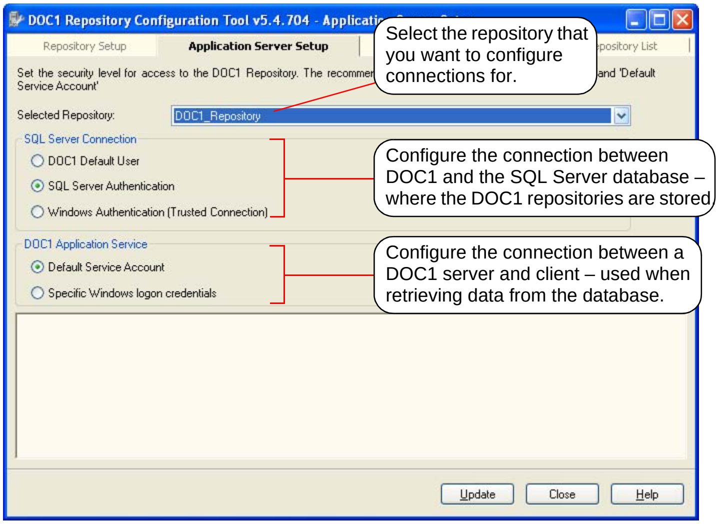
Filestore Backup File : C:\Program Files\Group1\DOC1\Designer\Backups\backup.zip

□ Do you want to compact the database (Not Recommended)?

Application Server setup

This allows you to configure the server security for connections both between a DOC1 server and client, and between your DOC1 environment and the SQL database.

▶ **To configure server connection:** select the *Application Server Setup* tab. Note that only repositories that are up-to-date can be configured. To upgrade a repository to the latest version see page 382.



Set the security level for access to the DOC1 Repository. The recommen and 'Default Service Account'

Selected Repository: DOC1_Repository

SQL Server Connection

○ DOC1 Default User

● SQL Server Authentication

○ Windows Authentication (Trusted Connection)

DOC1 Application Service

● Default Service Account

○ Specific Windows logon credentials

Select the repository that you want to configure connections for.

Configure the connection between DOC1 and the SQL Server database – where the DOC1 repositories are stored

Configure the connection between a DOC1 server and client – used when retrieving data from the database.

Application Server Setup

SQL Server Connection

DOC1 Default User – Uses the DOC1 default user and password, which is added to the configuration file.

SQL Server Authentication – This setting connects DOC1 to the SQL Server using a valid SQL Server user that has permissions for the SQL Server. This will be added to the configuration file.

Windows Authentication (Trusted Connection) – This setting enables DOC1 to connect to the SQL Server with Windows user credentials. This must be a valid Windows user that has permissions to access the SQL Server. You cannot use this option in connection with *Default Service Account* below.

> ⓘ Please note that if SQL Server is not running in *Mixed Authentication* mode, enter here the Windows user credentials under which SQL Server service is running.

DOC1 Application Service

Default Service Account – Sets the DOC1 service to run with the Network Service account that Windows uses.

Specific Windows logon credentials – Use a valid Windows user account that will be used to run the DOC1 service.

Importing and exporting publications and Active Content

The export feature allows you to capture the current state of an individual publication design and its associated resources, as well as public Active Content.

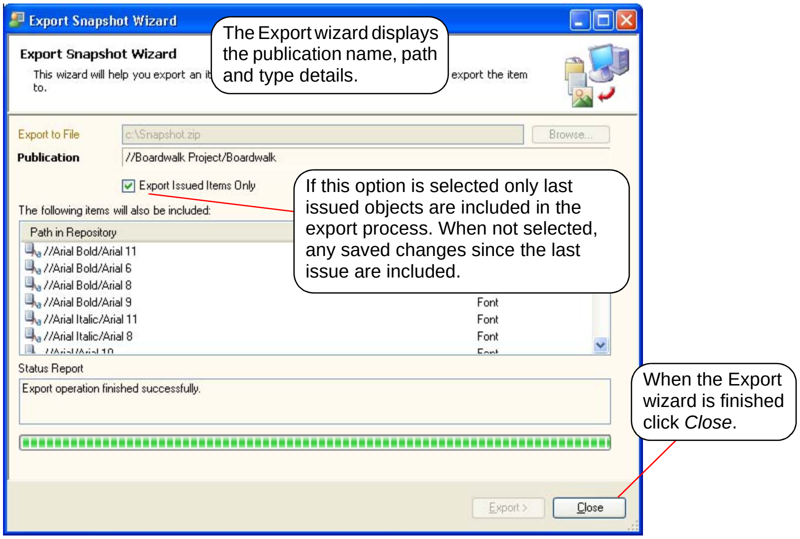
Together with the import feature it can be used for:

- copying a publication to another DOC1 repository, for instance when you need to transfer resources to DOC 1 Customer Support for analysis.
- basic version control. The import snapshot function in conjuction with repository backup and restore can be used to promote design resources across repositories whilst providing a means to return to a particular design version level.

When importing data dictionary objects that already exists in your repository, it is important to note that in order to preserve data integrity, deleted data fields and records are never removed from data dictionary.

▶ To export a publication:

1. In the Project Explorer open the required project and select the publication, or Active Content you want to export.
2. From the menu select Tasks/Export Snapshot. The items contained in the publication/Active Content are loaded and displayed in the Export wizard.



Export Snapshot Wizard

Export Snapshot Wizard

This wizard will help you export an it… export the item to.

Export to File: c:\Snapshot.zip

Publication: //Boardwalk Project/Boardwalk

☑ Export Issued Items Only

The following items will also be included:

```
Path in Repository
//Arial Bold/Arial 11
//Arial Bold/Arial 6
//Arial Bold/Arial 8
//Arial Bold/Arial 9          Font
//Arial Italic/Arial 11       Font
//Arial Italic/Arial 8        Font
//Arial/Arial 10              Font
```

Status Report

Export operation finished successfully.

The Export wizard displays the publication name, path and type details.

If this option is selected only last issued objects are included in the export process. When not selected, any saved changes since the last issue are included.

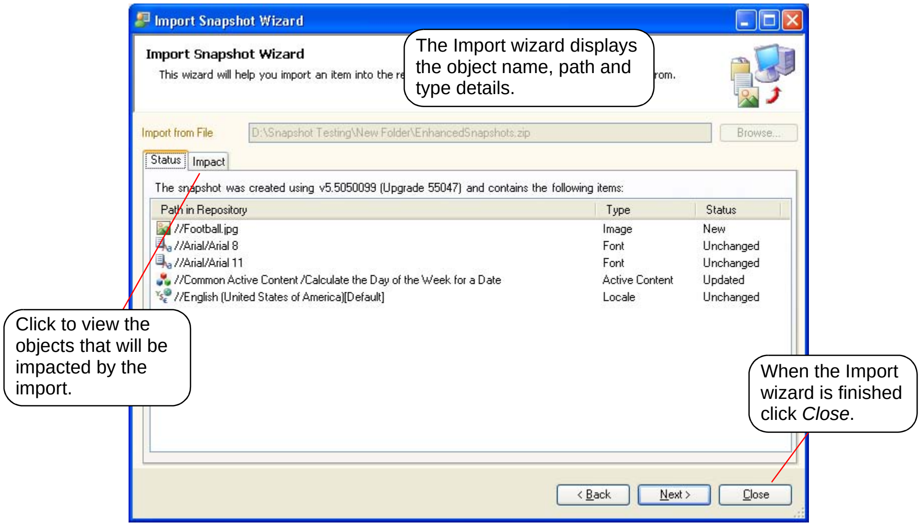
When the Export wizard is finished click *Close*.

3. Click *Browse* and use the *Select Export File* dialog box to specify a name and location for the snapshot.
4. In the Export wizard click *Export* to complete the process.

▶ To import a publication:

1. Double-click *Repository* in the Project Explorer and from the *Tasks* menu select *Import Snapshot*. The Import wizard is displayed.

2. Click *Browse* and use the *Select Import File* dialog box to search for the file. Click *Open* and the contents of the publication are loaded into the wizard.



Import Snapshot Wizard

Import Snapshot Wizard

This wizard will help you import an item into the re rom.

Import from File D:\Snapshot Testing\New Folder\EnhancedSnapshots.zip

The snapshot was created using v5.5050099 (Upgrade 55047) and contains the following items:

Path in Repository	Type	Status
//Football.jpg	Image	New
//Arial/Arial 8	Font	Unchanged
//Arial/Arial 11	Font	Unchanged
//Common Active Content /Calculate the Day of the Week for a Date	Active Content	Updated
//English (United States of America)[Default]	Locale	Unchanged

The Import wizard displays the object name, path and type details.

Click to view the objects that will be impacted by the import.

When the Import wizard is finished click *Close*.

Use the Impact tab to review objects affected by the import process, these will require republishing and retesting.

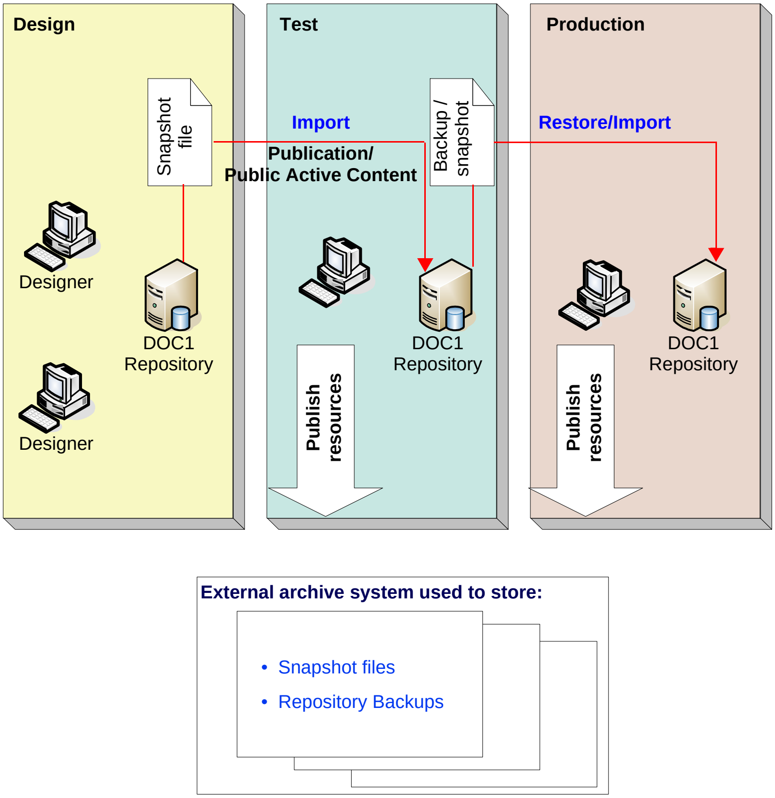
3. Click *Next* to start the import process.

4. The publication/Active Content and the project that contains it are placed in the location as defined in the Path in Repository field as shown in the screenshot.

Care must be taken when importing Active Content to ensure all dependant publications are published. If the Active Content is independently published then it also must be published.

Creating a managed environment for change control

If your installation requires greater control over DOC1 resources than is provided by DOC1 Designer itself, perhaps for compliance or other version control requirements, you should consider creating multiple DOC1 repositories and using snapshots to move application resources between them at key points in your development cycle. The diagram below illustrates a possible multiple repository scenario that caters for this type of requirement.



This shows a single design repository allowing for the creation and maintenance of document designs. Testing in the design environment is carried out either using Preview/Advanced preview or by creating HIP files. This repository is used as a source area from which resources are promoted to the other repositories for testing and carrying forward into production. Note that shared objects such as Public Active Content and images are recognized by the repository using an internal identifier rather than by name. For this reason it is recommended that a single source repository is used to avoid the creation of duplicate resources.

When a design is ready to be promoted for testing, the associated publication or Public Active Content is exported from the design repository as a snapshot file, which in turn is imported into the test repository. These resources can then be published to create a publication HIP for testing. Refer to “Importing and exporting publications and Active Content” on page 387 for details on working with snapshots. Note that any design changes identified as a result of testing must be made in design area and then promoted again to the test environment.

NOTE THAT YOU SHOULD ONLY EVER PUBLISH FROM ONE REPOSITORY INSTANCE TO ENSURE ALL THE RESOURCE MAPPING, OUTPUT DEVICES AND OTHER PRODUCTION SETTINGS ARE CONSISTENT IN THE ASSOCIATED HIP FILES.

The next step in the process is either to create a snapshot containing the relevant design resources, or to carry out a backup of the test repository. Before promoting these resources to the production repository a backup of the production repository is required. Refer to “DOC1 Repository Configuration” on page 381 for details on running the backup and restore functions. The snapshot/backup can now be imported/restored to the production repository and the updated resources published for production.

At the end of this process a set of snapshots and backups is created. These files should be stored in a version control environment in case there is a requirement to revert the design repository back to state under which a particular production HIP was created.

Appendices

Appendix A – Glossary

2of5

Type of barcode.

3of9

Type of barcode.

action

DOC1 functions that change the way an application is being processed at run time.

Active Content

Active Content files contain segments of document content and logic and provide a highly flexible way to share and reuse such segments across multiple publication designs.

ACU

This provides the ability to extract AFP codepage information within a HIP file.

address block

Design object that automatically formats name and address data into a style that is the standard for a named country.

Administration

The view where the administrator can create and maintain users and administer repository tasks.

Adobe font metrics

The font metrics used with PostScript fonts.

AFP (Advanced Function Presentation)

AFP is IBM's print and presentation architecture. Also the name sometimes given to a printer that this architecture supports.

AFPDS (Advanced Function Presentation Data Stream)

The output datastream of the AFP environment.

anchor

The method of position objects within a publication design.

anchored

Anchor method where graphics are positioned relative to paragraphs within the document design.

arithmetic

DOC1 function that performs calculations within publication designs.

ASCII

The character coding system used in Windows and UNIX operating systems.

audit trail

See Message audit trail

axes, axis

The lines on a chart that provide a scale references.

background

A design area whose contents are placed behind the main document layer. The background is often used for a watermark or similar.

bar

1. A section within the DOC1 Work Center Manager window that provides access to a group of objects.
2. An element on a bar chart

barcode

A code normally consisting of a series of vertical lines or bars of varying thickness which can be read by an optical scanner.

Base 14 fonts

A set of standard fonts that is available with most PostScript and PDF printers/ browsers.

bidirectional text

A combination of both left to right and right to left text in a paragraph.

bitmap or BMP

A computer image format that is the standard working format within the DOC1 Work Center. You can convert a wide range of other image formats to BMP.

body/body area

The primary area in a document design where content is added.

BookBill

A physical media layout that includes a payment slip or similar as an extension to the first physical page of a publication design.

bookmark

A DOC1 design object used for generating bookmarks within PDF output.

bounding box

A rectangular shape area that encompasses all elements of a presentation object.

browser

A program that allows the on-line viewing of documents such as Acrobat Reader for PDF files.

bundle

Design object that allows you to temporarily withdraw other objects from the design scope.

carriage return

See hard return.

case

Design object that selects one of a range of conditional paths when a comparison value is matched.

casting

Converting one data type to another, for instance, using a string containing 121205 as a date.

cell

The basic element of a table. Combines to make rows and columns.

centralized printer

The name sometimes given to a type of Xerox printer.

character set

One of three font files used in the IBM AFP architecture, a character set contains the actual raster patterns for the characters.

chart

A graphical representation of numeric application data. DOC1 supports bar, pie, line and scatter charts.

chart element

Used to represent a single numeric value within a chart, e.g. a single bar in a bar chart.

checkpoint

A means by which DOC1 Generate can report, at specific intervals, which publication data set is currently processing.

child object

An object that is within another object in a logic map.

code128

Type of barcode.

coded font

One of three font files used in the IBM AFP architecture, a coded font combines a character set and code page.

code page

One of three font files used in the IBM AFP architecture, a code page maps code points used in output data with actual characters in a character set.

column

1. Consists of any number of cells aligned vertically and is used with rows to create a table. Also used to specify the layout of a table on the page, for example, a three column table will split the table into three, side by side.
2. A rectangular section within the body area of a page for positioning main document content. Can have up to a maximum of eight. Defined using a column layout.

column layout

Used to define the layout of columns on a page.

composite resource

Any design object that contains other design resources, for instance a publication that contains document designs.

concatenation

Joining of two or more text values together to form a single string.

condition

Design object that queries an expression and executes either a *True* or *False* logic map path depending on the result.

conditional paragraph

A paragraph that has criteria associated with it defining when it should be included in the document output.

constant

Provides non-variable data to objects.

Content Author

Content Author is an optional module of the DOC1 environment that allows key parts of publication content – messages – to be created and maintained independently of the main publication design.

control object

The data element that is used within a repeating data object.

currency

A DOC1 data type used for monetary values.

Data

The DOC1 Work Center Manager tab that gives access to the data format editor, data dictionary and data maps.

data alias

A reference to a field or record in a publication design. Data aliases enable you to create a publication design without having sample data available.

database

This is the part of the DOC1 Work Center repository that provides revision control.

data dictionary

A group of existing data aliases.

data field

A field from input data used in a publication design. This is an alias to the actual field and is mapped to the appropriate data element before the publication is put into production.

data format & Data Format Editor

A data format describes the type and structure of the input data intended to be used with a particular DOC1 job. It is created using the Data Format Editor (DFE) in the DOC1 Work Center.

data map

Used to map data aliases from a particular data dictionary to real data in a data format.

data input user exit

A type of user exit that takes control over locating, parsing and delivering data to DOC1 Generate.

data record

A record from input data used in a publication design. This is an alias to the actual record and is mapped to the appropriate data element before the publication is put into production.

date

A DOC1 data type for calendar dates.

DBCS

Double byte character set. This term is used to describe fonts that have more than 255 characters and which therefore cannot be uniquely identified with a single byte – typically the case with Chinese, Japanese, Korean and other pictorial languages.

default page header/footer

Used to define the height of headers and footers in the current document and to format how these areas appear on the page.

deployment

The process of sending publication resources to a production system for subsequent processing by DOC1 Generate.

delimited data

A structure of input data supported by DOC1. This is a variation of keyed record data where field locations are automatically identifiable to DOC1 by the presence of one or more consistent delimiter characters between each field.

Design

The DOC1 Work Center Manager tab that gives access to publications designs and their sub-components (documents etc.).

DFE

See Data Format Editor.

dictionaries

1. The reference files used by the spell checking and automatic hyphenation features.
2. See data dictionary.

DIJ – DOC1 interchange journal

Provides an index of documents within an output datastream for use with e2 Vault and e-Messaging.

DIME

Dynamic Internet Mailing Engine. This DOC1 utility is provided to extract the individual documents from a PDF stream generated by DOC1 Generate and, optionally, to e-mail them.

DJDE

Controls stored within a Xerox Metacode datastream. They can specify dynamic changes to the Xerox print environment.

DOC1ACU

See ACU (AFP Codepage Utility).

DOC1DIME

See DIME.

DOC1GEN

See Generate.

DOC1RPU

See RPU (Resource Pack Utility).

document

The basic design unit of a publication. Typically, publications have only a single document design which is made available automatically when the publication is created. However, you can create additional document designs for conditional selection or reuse within a range of similar publications.

document area

A logical area that defines the dimensions within which messages are designed in Message1. Messages associated with a document area can appear multiple times in the same document.

Document Editor

The main DOC1 document layout tool.

document section

See section.

dot leader

A row of dots between the end of preceding text and a tab stop, typical of table of contents and similar styles.

Double byte fonts

See DBCS.

draft

The status of a DOC1 design resource before it is issued.

dynamic job descriptor entry

See DJDE.

e2 Vault

See Vault

EBCDIC

The character coding system used on IBM z/OS and OS/400 operating systems.

embedded resources

Fonts and image resources that are actually contained within an output datastream rather than being referenced.

e-Messaging

The e-Messaging system manages e-mail and SMS (Short Message Service) customer communications.

EMFE (Extract Merge and Format Engine)

The main production engine program of the DOC1 Series 4 (and earlier) environment.

EngageOne

EngageOne is an interactive document production system, which enables the creation and delivery of ad hoc customer correspondence.

Environment

The DOC1 Work Center Manager tab where you can configure hosts, output devices, locales etc.

EPS

Encapsulated PostScript file that can be imported into a publication.

eye-catcher

A sequence of bytes used to provide a human-readable reference in input data records.

external document

A pre-composed document in a supported external format that can be inserted into a publication.

field

General term used for the various values that can be used in a document, including data field, variable and system variable.

filestore

This is the part of the DOC1 Work Center repository that contains DOC1 files and other resources used when designing publications.

fixed anchor

Anchor method where a graphic is positioned at a fixed offset on the page where it is inserted (or multiple pages if used on the background).

font

A set of printable or displayable text characters in a specific style. In DOC1 you design publications using Windows fonts which are then converted (if necessary) to a format suitable for your intended output device.

font subsetting

Only those characters that are used in a document are embedded in it. This reduces the size of the output file produced by DOC1 Generate, particularly when DBCS fonts are used.

footer

An area at the bottom of a page or table containing any type of content. Can be specified to appear only in certain conditions, for example, only on the first page.

form

The Xerox term for an overlay.

formatting marks

Non-printing characters that indicate controls within paragraphs such as end markers, spaces, tab characters and so on.

formula

Can be used as part of some design objects to specify more complex parameters than can be provided by dialog options.

Generate

The DOC1 batch program that processes production jobs on your chosen host system. It reads information about the job requirements from a DOC1 HIP file, merges the input data file it receives with your publication designs and produces output datastreams ready for printing or presenting on your intended output devices.

governor field

Indicates the number of times data is repeated to a repeated data object.

hard return

Used to create a new paragraph by pressing the ENTER key.

header

An area at the top of a page or table containing any type of content. Can be specified to appear only in certain conditions, for example, only on the first page.

HIP

Created by a publishing task, a HIP file contains all the information and resources required for a particular publication in the production environment. It also typically contains the resources required to print/present the resulting output datastream on your intended output devices.

host

Provides properties for the computer where the Generate program will be run.

HTML

Hypertext Mark-Up Language is used for displaying information on the web.

IJPDS

AN output format that is designed to be printed on the Scitex family of high speed ink jet printers.

image

A graphics file used by design files.

indentation

The position of a paragraph in relation to the page margins.

indexes

The references used to assign data to chart elements in some types of charts.

in-line anchor

Anchor method that positions graphics as a character within paragraph text.

input data

The data file that is provided as input to a DOC1 job in the production environment.

insertion point

In the logic map, this indicates the object after which the next object will be added.

integer

A DOC1 data type. An integer with up to 9 digits.

issue

1. A status of a design object within the design repository. An issued object is currently active and available for use when publishing a publication design.
2. A revision of a design object within the repository.

Interactive Data Editor (IDE)

The IDE is a Windows application, which allows prompts, help text and other attributes (metadata) of interactive data fields to be defined for EngageOne.

JCL (job control language)

Controls specified by the user to define and execute jobs under z/OS.

JDL (Job descriptor library)

A library containing compiled job descriptions for Xerox Metacode printers.

journal

A file that records the activity of a publication when it processes in the production environment.

JSL (Job Source Library)

A library containing uncompiled job descriptions for Xerox Metacode printers.

keycodes

The codes used to protect DOC1 software from unauthorized use. Various keycodes will be issued to the user depending on their software and hardware configuration.

key field

In keyed record data this is the field that uniquely identifies each record type.

key map

Maps keys for keyed objects to the actual image or Active Content when using resources that are in the Work Center.
If images are supplied externally, it maps keys to sample images for use when designing, with the actual images being provided via an external control.

keyed object

Keyed objects provide a means of allowing content (Active Content) and images to be selected dynamically at production time based on variable criteria or 'keys'.

keyed record data

A structure of input data supported by DOC1. This is effectively raw data although it must have a recognizable record structure. Each record type must be uniquely identifiable by the contents of a key field.

keyed user exit

Returns the image to be placed as a keyed object.

legend

The key to the elements shown in a chart.

lines

1. Drawing objects that can be used in document designs.
2. The lines that connects two data markers in a line chart
3. Optional rules on some chart types that provides an additional scale reference.

line data

Organized text data with an implied structure that can be used by suitable input/output software but which does not contain any actual controls within the data itself.

locale

DOC1 object that provides settings related to the customization of data and presentation objects to take account of regional variations.

lock/unlock

In the DOC1 design repository resources that are being edited are automatically locked. They are normally unlocked automatically but the user can choose to keep the lock when the resource is not open.

logic map

An icon based view of the instructions you are generating as you add objects to a design.

logical page

DOC1 Generate produces logical pages within output datastreams. You can configure a job to place multiple logical pages on a single physical page.

logo

A format of graphic used on Xerox printers where the image is stored as one or more font characters.

lookup table

Keywords passed to a lookup table are replaced with the appropriate text string if a match is found.

lookup table user exit

A user exit that overrides any existing lookup table functionality allowing an application to access reference tables not normally supported by DOC1.

loop

Design object that queries an expression and continuously executes a logic map path until it is no longer *True* or until a *Break Loop* is encountered.

mandatory records

Records in input data that must be included in every publication data set.

margin

The space around the edge of a page that is not available as part of the body area.

message

An object created and controlled by Message1 or Content Author.

message area

A logical area that defines the dimensions within which messages are designed in the message applications.

messages file

The file (.HIM) containing Message1 or Content Author objects associated with a particular campaign. It contains all the information and resources required for messages in a production run.

message box

Reserves fixed space in a document for messages from a message application environment.

message stream

Allows messages from Content Author to be inserted directly into the flow of the page in the DOC1 publication.

Message1

Message1 is an optional addition to DOC1 that allows key parts of a document – messages – to be created and maintained outside the Work Center.

Message audit trail

This is similar to a regular DOC1 journal file but is specifically and automatically used to log messages that have been included within documents.

message environment file

This contains information about the available message areas and message data fields. It is an XML file generated in the message application and is imported into the Work Center.

metacode

The output datastream used with Xerox Centralized printers. Also the name sometimes used for printers using this datastream.

MVS

The previous name used for IBM's z/OS mainframe operating system.

non-repeating data

Fields that typically appear only once in the input data for a single publication.

note

An object that allows you to provide commentary within a publication design.

number

A DOC1 data type, this is a decimal number with a maximum of 24 digits before the decimal place and 8 after.

OpenType font

The OpenType font format is an extension of the TrueType font format, adding support for PostScript font data.

OPS (override production settings)

Used in the DOC1 production environment, an OPS file allows you to specify additional or alternative file references and other settings to those use when publishing a publication design.

orphan

A few lines at the end of a paragraph that are separated from the rest of the paragraph by a page break.

output datastream

The main output of DOC1 applications. An output datastream contains the composed pages in the low-level format suitable for your target printer/browser.

output device

In the DOC1 Work Center an output device object is used to indicate the output datastream required by a particular job plus any device specific settings you may need.

outsorting

Splits the output from a single production job into multiple stream to support the needs of post-printing equipment such as enveloping machines or similar.

overlay

An overlay is the term used by IBM for an electronic form. The Xerox equivalent of an overlay is a form (FRM) file.

owner password

Protects the attributes assigned to a PDF file from unauthorized modification.

page check

A specialized version of a condition that allows you to query the current page number within a document or section to see if it matches those specified.

page setup

A document design function that defines the page size, headers, footers, background and new page logic.

paragraph

A continuos piece of text which can include graphics, images, text boxes anchored to it. Has properties such as alignment, spacing, tabs.

parameter

A value passed into Active Content.

PCE

The DOC1 Post Composition Engine (PCE) handles the requirements of some applications for additional processing once document pages have been composed.

PCL

A control language used to control printers or other output devices based on the Hewlett Packard family of laser printers.

PDF

Portable Document Format is a file based document format suitable for Internet distribution and on-line viewing with a PDF viewer such as Adobe Acrobat Reader.

pel (picture element)

In printing, the smallest area that can be individually toned by the printer.

physical page

An actual piece of paper on a printer.

Post Composition Engine

See PCE.

PostNet

A type of barcode.

Postscript

A common page description language acceptable to many desktop printers and also, increasingly, for high volume production printers.

PPML

Personalized Print Markup Language. This is an XML based page description language developed for high speed output datastream generation.

preview

Runs the current publication through DOC1 Generate and displays the resulting output into a window. This can then be sent to the local printer.

production job

The production settings associated with a publication as saved in the DOC1 Work Center.

project

Can be used to group design resources for organizational and access control purposes.

public

Property of some design resources indicating that they can be shared by multiple publications.

publication

An assembly of one or more related documents.

publication data set

The input data intended for an individual publication.

Publication Editor

Where you specify the documents that make up a publication and define publication properties.

publish

A task that prepares a DOC1 job for the production environment. It creates the HIP file used by DOC1 Generate to control processing.

RDW (record descriptor word)

A method of delineating records in input data.

record

A segment of input data. A record is typically a line of data in the input file containing one or more fields.

repeating data

A data element that is iterated within a single publication data set. Repeating data typically contains transaction type information

repository

This stores and provides revision control for all the design resources and other objects that are used in the DOC1 Work Center. These include resources that are created within the Work Center itself or have been imported into it from an external source. The repository consists of a filestore and a database.

Repository Configuration Tool

A stand-alone tool for configuring and managing repositories

reserved area

Only available on the background, it defines space on the page that cannot be overwritten by content within the main document body which will flow around the reserved area as necessary.

reserved area entry

A means of dynamically updating the content of a reserved area.

resource map

A tailored mapping of resources created by a publishing task. A resource map overrides the default names given to the font and image files (and the references used to identify the resources within an output datastream).

revert

This function allows you to restore to an earlier revision of a non-composite design resource.

revision

A version of a design resource.

RPU (Resource Pack Utility)

This provides the ability to extract and manipulate resources within a HIP file.

RTF (Rich Text Format)

A document mark-up language commonly used to exchange data and graphics between different applications. Can be imported into documents and is an output format supported by Generate.

rollback

This function allows you to restore to an earlier revision of a composite design resource.

row

Cells aligned horizontally within a table.

rulers

Measurement guides that can be shown at the top and side of the Document Editor.

sample data

An extract from intended job input data. This is used as a template for defining a data format.

scope

The level at which variable names are reserved. For example, a value in a variable with *Publication* scope is available throughout an entire publication.

section

A sub-set of logic map objects within a document design. You can reuse sections within the same document.

segmented publishing

A term used where the publication design is published separately from the output device configuration and resources. Any number of separate resource HIP files can be produced for use with DOC1 Generate.

set

Provides categorization in some chart types.

shape

Lines, rectangles, rounded rectangles and ovals that can be drawn in a document.

sheet

An actual media page as used on a printer or produced by an electronic document format. A sheet is the area on which the logical pages produced by DOC1 are placed.

SMS (Short Message Service

A means of sending text messages to and from mobile phones.

soft return

Used to create a new line within the same paragraph by pressing the SHIFT + ENTER keys.

sort

In DOC1 terms, the reordering of repeating data prior to its use with a publication design.

spell checking

Verification of paragraph text using the dictionary of the assigned language.

start of publication record

Data format property that identifies the data element that indicates the start of a new publication data set.

static table

A regular table in a document design. It cannot contain repeating data.

string

A DOC1 data type that can contain any alphanumeric characters can be used up to a limit of 32k characters.

substring

The substring function extracts one or more consecutive characters from within a text string.

symbol

A character that cannot be accessed directly from the keyboard.

system variable

Information from the production environment that can be included in publication designs, for example current page number or date.

table

Aligns text and graphics in rows and columns or generally helps with positioning objects on a page (as with most word processing tools). Can be a static table or a transaction table.

tag logical element (TLE)

An AFPDS structured field that is the functional part of an AFP index. TLEs can form part of an independent index file or be included directly in the composed pages to which they relate.

template

The package that contains the publication and the interactive sections of a document, ready for use by the correspondent in EngageOne.

text box

Acts a container for specific page content and can contain both paragraph text and graphics. Can be used to highlight particular content or as a reserved area which other paragraph text can flow around.

time

A DOC1 data type for time values.

transaction table

Controls how repeating data is presented, i.e. how multiple iterations of the same repeating data structure are ordered and how they are positioned in relation to one another.

TrueType font

An Windows outline font where the characters are scalable.

unlock

See lock/unlock.

URL

Uniform Resource Locator – the global address of a document or file on the Internet.

user

An individual given a user name and password, enabling them to work in the DOC1 environment.

user exit

DOC1 user exits allow an application to call a user defined program function. DOC1 supports user exit programs coded in either C. There are four types of user exit: data input, file, lookup table and keyed.

user group

A list of users who share similar responsibilities. Groups are used for logical groupings of multiple users and enable the administrator to manage a number of users at one time.

user password

Used to protect PDF documents from unauthorized opening in a PDF browser.

variable

Variables store values for subsequent use. They can hold values of any supported data type, but each individual variable can only accept data of a single type.

value

Values in document designs can be provided by data fields, variables, system variables, the output from functions or you can simply enter a constant.

Vault

The e2 Vault environment caters for high speed storage and retrieval of electronic documents. e2 Vault itself is the main repository in which documents are stored and maintained. Various other e2 products provide access to the repository.

vector offset

A reference to the start of individual composed documents within DOC1 job output.

VIPP

Variable Data Intelligent PostScript PrintWare. This is a PostScript based page description language developed by Xerox for use with VIPP enabled printers.

VPS

Variable Print Specification. This is a PostScript based page description language developed by Scitex for use with VPS enabled printers.

VSAM

A data set structure used under z/OS.

widow

A few lines at the beginning of a paragraph that are separated from the rest of the paragraph by a page break.

XML

XML – Extensible Markup Language – provides a common method of describing data for internet and other applications. DOC1 supports XML as a format for input data.

zoom

Magnification used in an editor.

Appendix B – Formula expressions

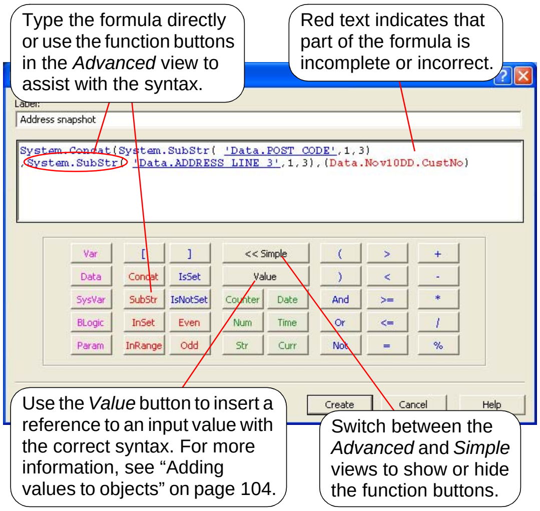
Formulas can be used as part of some design objects to specify parameters more complex than can be provided by simple value controls or as a more convenient way of coding compound functions. For instance, you could create a complex statement for a condition such as:

```
SubStr(CN,0,1) = "V" And Pay > 35000
```

Or you might use a formula to code a calculation rather than using nested arithmetic objects:

```
(GrossPay - Deduction) * TaxRate
```

In the *Advanced* view of the *Formula* dialog box you can use the various function buttons to help with coding the syntax.



Label: Address snapshot

```
System.Concat(System.SubStr( 'Data.POST CODE',1,3)
,System.SubStr( 'Data.ADDRESS LINE 3',1,3),(Data.Nov10DD.CustNo)
```

Type the formula directly or use the function buttons in the *Advanced* view to assist with the syntax.

Red text indicates that part of the formula is incomplete or incorrect.

Use the *Value* button to insert a reference to an input value with the correct syntax. For more information, see “Adding values to objects” on page 104.

Switch between the *Advanced* and *Simple* views to show or hide the function buttons.

Formula syntax

A Formula expression consists of a single statement. Compound statements are allowed but assignments and program logic are not supported. All keywords plus data and variable names are case sensitive.

Objects and functions are referenced using their name plus their type and a varying number of other qualifiers. You may exclude some or all qualifiers (always working left-to-right) where there is no possibility of duplication or ambiguity. For a data field reference for example, you may have a fully qualified name of:

```
Data.MainDic.CustomerAddress.CityName
```

Provided 'CityName' is not used as a field or variable name elsewhere in the publication design you may simply code:

```
CityName
```

If the field name is used in another record of the same data dictionary you can code:
`Data.MainDic.CityName`
You may not code:
`CustomerAddress.CityName`

If a name contains spaces or characters used as part of the formula syntax itself the entire name must be contained within single quotes. For example:
`'Var.Pub.Current Rate'`

Constants

Defines a value and an associated data type.

Counter	an integer `362436`
Currency	`C'$9.99' or Currency'$9.99'`
Date	`D'13/10/2003' or Date'13/10/2003'`
Numbers	`18.67`
Strings	`"Account balance"`
Time	`T'11:22:33' or Time'11:22:33'`

Data Fields

`Data.`*`Dictionary`*`.`*`Record`*`.[`*`Record`*`...]` *`Field`*

Returns the value of *Field*. *Dictionary* indicates the name of a data dictionary object. *Record* and *Field* are data elements within the dictionary. You may need to specify multiple record names where these are nested within the dictionary. All qualifiers other than *Field* are optional.

Examples:

```
Var.Temp=Data.AnexDic.R1.R12.Taxes
Currency(Data.Taxes)
IsNotSet Ref001Taxes
```

System Variables

`SysVar.`*`Name`*

Returns the current value of a DOC1 system variable. *Name* is one of:

```
RunDate
RunTime
GenerationDate
GenerationTime
PublicationSequenceNumber
DocumentPageNumber
PublicationPageNumber
JobPageNumber
```

See “System variables” on page 155 for details.

Examples:

```
Var.Temp=SysVar.JobPageNumber
SubStr(RunDate,1,4)
```

Variables

`Var.`*`Scope`*`.`*`Type`*`.`*`Name`*` [`*`index`*`]`

Returns the current value held in variable Name. If *Name* is an array *index* must be specified to indicate the element required.
Scope and *Type* may be used to identify the variable uniquely if duplication is possible:
Scope – indicates the defined scope of the variable.
Options are `Job, Pub, Doc` or `Loc`.
Type – indicates the data type of the variable.
Options are `Counter, Num, Str, Date, Time` or `Currency.`

Examples:

```
Var.Temp=Var.Job.Str.Addr[2]
Date(CDateString)
IsSet Var.Time.Tdate
```

Conditions

Comparison operators: When using a formula within a condition object the expression is evaluated using one of the following operators and TRUE or FALSE is returned.

`>`	Greater than
`<`	Less than
`>=`	Greater than or equal to
`<=`	Less than or equal to
`=`	Equal to
`<>`	Not equal to
`IsSet `*`object`*	returns TRUE if *object* is a data field that is available in the current data set or is a variable currently in scope.
`IsNotSet `*`object`*	returns TRUE if *object* is a data field that is unavailable in the current data set or is a variable that is not currently in scope.

Examples:

```
Var.DayCode = 7
Data.Received > D'13/10/2003'
Var.Option <> ¨Gold¨
IsNotSet Var.CountR
```

Logical Operators: You may also use these operators within conditions to join expressions and return the result. Note that any function that returns TRUE as its result (such as InRange, InSet, Even, Odd) must specifically test for "= TRUE" when the function is used as part of a formula using the logical operators.

AND	returns TRUE if both expressions are TRUE
OR	returns TRUE if either expression is TRUE
NOT	returns TRUE if an expression is FALSE.

Examples:

```
Data.Tax > 10 AND IsNotSet Var.CountR
(System.InSet(Var.Code,312,315,317) = TRUE) OR
(System.InRange(Var.Code,320,350) = TRUE)
```

Functional Operators

Use these operators to perform calculations of input values. You may calculate dates within the restrictions of the regular DOC1 arithmetic function. See “Date calculations” on page 145 for details.

+	addition
-	subtraction
*	multiplication
/	division
%	modulo (remainder)

Examples:
Var.Job.Str.Addr[2] + 10
SysVar.RunDate –¨2M¨

Functions

The following functions can act on values supplied to a formula and return a value. Use the qualifier *System* if necessary to ensure their name is uniquely identified.

Concat

Concatenate multiple strings.

```
System.Concat(string1, string2...)
```

Returns the input strings as a single concatenated string.

Example:

```
Concat(¨Today is ¨,SysVar.RunDate)
System.Concat(Var.T1, Var.T2, Var.T3)
```

Even

```
System.Even(counter)
```

Returns TRUE if *counter* is even.

Example:

```
Even(Var.MonthCode)
```

InRange

Search a range of values for a match.

```
System.InRange(target, low, high)
```

Returns TRUE if the *target* is between *low* and *high*. Any data type can be used with InRange except strings but all values must be of the same type.

Example:

```
InRange(Var.Tcode,1,20)
System.InRange(Rating,5,MaxRating)
```

InSet

Search input values for a match.

```
System.InSet(input, comp1, comp2 ...)
```

Returns TRUE if *input* matches any of the comparison values. You can have as many comparison values as necessary. All parameters must be of the same data type.

Example:

```
InSet(Var.Tcode,1,2,7,10)
System.InSet(Rating,6,9,MaxRating)
```

Odd

`System.Odd(`*counter*`)`
Returns as TRUE if *counter* is odd.

Example:
`Odd(Var.Pub.Counter.Weekday)`

SubStr

Substring
`System.SubStr(`*source*`, `*offset*`, `*length*`)`
Returns a substring of string *source* starting at position *offset* and continuing for *length* bytes.

Example:
`SubStr(Var.FullAddress,1,20)`
`System.SubStr(RunDate,6,8)`

Casting

`System.`*type* `(`*input*`)`

Returns *input* converted to the specified data *type* where possible. *Type* can be:
Currency
Date
Counter – integer, a whole number up to 9 digits
Num – number
Str – string
Time
Note that, wherever possible, DOC1 will automatically convert data types to meet the requirements of a function. You need only code explicit casting where there is ambiguity. The format of the return value (from both explicit and implicit casting) is based on the DOC1 locale that is active when the function is called. See "About locales" on page 183 for more information. Use the qualifier *System* if necessary to ensure the function is uniquely identified.

Examples:
`System.Date(¨06/07/2005¨)`
`Num(Var.TCode)`

Active Content

Blogic

Call Active Content function.
`BLogic.`*Name* `(`*parm1*`, `*parm2*`, …)`
Returns the return value of Active Content function *Name* which is called with any required parameters. See "Using Active Content" on page 224 for more information.

Example:
`BLogic.GetRate(3) = Var.MaxRating`

Param

Accesses Active Content parameter.
`Param.`*Name*
Use only within an Active Content function itself. Returns the value of parameter *Name* which must be a parameter passed to the function.

Example:
`InRange(Param.Rating,1,5)`

Appendix C – DOC1 Interchange Journals

A DOC1 Interchange Journal (DIJ) is an XML file that records information about the documents within a DOC1 generated output datastream. Each record within a DIJ specifies a unique key for each document, such as an account number. This can be used by other products, such as e2 Vault or e-Messaging, as an index into the output datastream. Note that each product will make use of a different set of DIJ options.

The DIJ data is created as part of the attributes of a publication.

▶ To create a DIJ:

1. In the Document view, from the *View* menu select *Publication*
   – or –
   right-click in the logic map and select *View Publication* from the shortcut menu.
2. Double-click the Publication icon in the logic map and the *Publication* dialog box is displayed. Select the *Journals* tab.
3. Select *Create DOC1 Interchange Journal* and click *Edit DIJ*. The *Document Interchange Journal* dialog box is displayed.
4. Supply values for the required parameters.
   Note that if you wish to use the *Address* or *Pay Details* tabs you must set parameters in the *Options* tab first.

ALL VALUES ARE ADDED USING THE REGULAR DOCUMENT EDITOR MECHANISM. SEE “WORKING WITH VALUES” ON PAGE 104 FOR DETAILS.

DIJ dialog box

Data

These parameters are used to identify individual publications.

Account Number — A unique reference number for the intended recipient of the publication (usually an account number).

Name — A name that uniquely identifies the publication.

Statement Date — A date associated with the publication, such as creation date, date of usage.

(continued)

DIJ dialog box

Publication ID – for use by e2 Suite

These options uniquely identify the company (vendor) and type of publication to which documents relate. Unless specifically required, use the *Default* setting. Options are:

Default
The references are automatically created. If you use this setting and are working with documents that relate to multiple vendors you must ensure the publication is stored within a project otherwise the vendor ID generated will not be unique.

Specify
Selecting this option allows you to enter the ID references manually. You will need to supply data that uniquely identifies each category:
Vendor ID – a reference that identifies a specific company;
Publication ID – a reference that uniquely identifies the type of publication;
Master ID – any additional unique references required for the application.

Read from .PSE file
Use this option if the relevant ID settings are available elsewhere. This will enable a *Locate* option where you can browse for the file.

Options

You can specify which parameters will be available in other tabs in the DIJ dialog box.

Skipped Pages
If your publication design produces banner pages that you do not want to be included use this parameter to indicate the page numbers that can be skipped in each document processed.

Additional street address lines (0–5)
If you want to include more than one line of main address information in the index use this option to set the number of additional lines required. The address is specified in the *Address* tab.

City, State/Region, Zip/Post Code, Country, Phone
Use these options to enable additional address fields in the *Address* tab.

Payable
This enables the parameters in the *Pay Details* tab – for use by e2 Suite.

Use Secondary Currency
This option enables a second currency in the *Pay Details* tab if required – for use by e2 Suite.

Store Data Set
Use this option to include the entire data set for each publication within the DIJ – for use by e2 Suite.

Address

Use this tab to provide the address data related to the account to which each document belongs.
The options available in this tab are specified in the *Options* tab.

(continued)

DIJ dialog box

Pay Details – for use by e2 Suite

These parameters are enabled by the *Payable* and *Use Secondary Currency* settings in the *Options* tab.

Due Date
The date that the bill being generated must be paid by

Primary Currency
A group of parameters:
Currency – the currency in which the amount values are expressed. The currency parameter must be resolved with a three character BSI Currency Code (ISO 4217 Maintenance Agency).
Due – the total amount due.
Min Due – the minimum amount due if partial payment is acceptable.

Secondary Currency
If your application supports two payment currencies use this parameter group to specify the currency and values of the secondary currency type. Parameters are as for *Primary Currency*.

Custom Settings

You can use custom settings to pass additional document information that may be required. Each entry provides you with a name and a value field for you to enter a user specified value to be included in the DIJ.

You must specify that the DIJ file is to be generated when the document is published.

▶ **To generate a DIJ:** in the Publish wizard *Output Device* page, select the *Configure DIJ for e2 or e-Messaging* option. This will enable you to specify the file name and path for the DIJ.

ⓘ See “Output Device” on page 340 for more details.

Index