



USER GUIDE AND REFERENCE MANUAL

# **Xara Online Web Service Modules**

---

© Xara Ltd 2002

Gaddesden Place • Hemel Hempstead • Herts • HP2 6EX

Phone 01442 350000 • Fax 01442 350010

Xara.com

First Edition 20<sup>th</sup> December 2002

# Table of Contents

<b>C H A P T E R 1</b>		Search your database and display formatted results	23
Introduction to Xara Modules	1	Automatic updating of a database listing	23
Benefits summary	3	Edit a record	24
Example applications	3	Update your website	24
Terminology	4	Step2: Sketch your Forms	24
The Module Store	6	Step 3: Create your Modules	25
Products	7	Step 4: Test the Modules and Processes	26
Usage Quotas	8	Web Page redirect testing	27
		Testing database writing	27
		Testing Reporter Modules	27
		Step 5: Integrate the Module into your website	28
<b>C H A P T E R 2</b>		<b>M O D U L E R E F E R E N C E</b>	
Xara Modules Architecture	9	Form Module	29
Dataflow Architecture	9	Mailer Module	31
Inputs and Outputs	10	Web 2 SMS	33
Fieldnames and Records	11	SMS 2 Web	35
Applications and Processes	12	Database Module	36
Data Types	13	Database Reader	38
XML Web Services	15	Database Writer	39
		Reporter Module	41
<b>C H A P T E R 3</b>			
Web Pages and Redirects	16		
<b>C H A P T E R 4</b>			
How to Design your Application	18		
Which modules?	18		
Database Module			
Database Reader Module			
Database Writer Module	19		
Database Synchronization	21		
Suggested steps to design your app	21		
Step1: Sketch your schematic	21		
Emailing Form contents	22		
Storing form data in a database	22		
Storing data and sending an email	22		
Search your database	23		

---



# Introduction to Xara Modules

## *In this Chapter:*

- Introduction
- Benefits
- Example Applications. What you can build.
- Terminology
- The Module Store
- Products
- Quotas

## Introduction

The Xara Connectable Module system is a revolutionary way to create web-based applications. For perhaps the first time, non-programmers can now create their own customized, database-driven applications and embed these easily into any website.

So what's the benefit of the Xara Module approach? That's easy. A ten times increase in the speed with which online data applications can be created. Put another way this could be a ten times reduction in the cost of developing such applications. And of course because it doesn't involve any programming it's usable by a far wider audience than other traditional methods.

Sceptical? Users of the Xara Modules system have created, from scratch, Photo Gallery applications in just two hours, simple content management solutions in the same time. Simple database applications where, for example, data is collected from a website and stored in a database can be created in minutes. At the other extreme large and complex applications, such as the Xara Membership Manager, which developed any other way would require thousands of lines of code, and many months to develop, were created in a few man-days of technical work.

The Xara Modules approach is dramatically more productive because it provides the component parts to build a huge range of different web-based applications. You can customize the application as you require and you can integrate it seamlessly into any website.

Building your own application is not difficult. It's a bit like LEGO® – you need to do a bit of planning before you build your finished product. As with most things, the more you plan ahead, the easier and quicker it is to build the right finished product.

Xara Modules is a 'no-programming' system. Its power and flexibility comes from being able to configure each Module using its web-based Editor and being able to connect the different Modules in different ways. If you can use typical desktop applications such as Microsoft Word or Excel, and certainly if you can use more complex products such as Microsoft Access, you can build your own customized web-based application using Xara Modules.

Better yet, you can try the Modules, for a limited trial period, free of charge.

Xara Modules is a hosted solution – we host the Modules on our servers that make up the web application that integrates into your website. This has many advantages not least being considerable ease and simplicity of creating these applications and not having to worry about database hosting, server configurations and so on.

## Benefits summary

- Quick, easy, no-programming approach to web application development
- Entirely web-based. Nothing to download or install on your desktop
- Hosted solution. Nothing special is required on your servers
- Compatible with all web authoring tools and all websites no matter where or how they are hosted
- Backend Database software included. No database licenses required and no database software required on your servers
- Applications fully, seamlessly, integratable into any website
- Safe, secure, reliable. We guarantee it. Multiple redundant hardware. 99.99% uptime with multiple backup strategies and continuous 24x7 monitoring.
- Easy data synchronisation. Direct from your desktop you can synchronise with your web database to, say, update your web database or download the latest website submissions.
- Highly cost effective

## Example applications

There is an almost unlimited range of web applications that can be created by Xara Modules. Just about application that requires a back-end database, where users can interact with this data – adding, editing and searching should be easy to create.

Using the Mailer and SMS Modules you can incorporate email and SMS texting functionality. From something as simple as having form contents emailed to being able to update or query your data via your mobile or cell phone. You can even make your website updateable from your phone.

A wide range of content management application can be created using the Reporter Module. At the simplest level this might be allowing easy updating of sections of your website to more advanced data or content management solutions.

Password protected data, web pages and applications are all possible.

Using the SmartPhoto Module you can build a range of photo supporting applications such as photo galleries, or product databases.

## Terminology

This is a description of the terminology used throughout this guide.

### WEB APPLICATION

---

A web-based program running on a web server and controlled and accessed via a web browser from the Internet. Anything that involves a form on a website will always have, albeit sometimes a very simple, web application running on a server. Hotmail, Google and shopping cart websites are common examples of more complex web applications.

### MODULE CONNECTABLE MODULE XARA MODULE

---

Applications are built from components. We call each component a **Xara Module** or, more specifically, a **Connectable Module**. (In this document we use **Module** as shorthand for a Xara Module or Connectable Module.)

Xara provides over 40 types of Module but only the newer Connectable Modules can be connected together to create a customized Web Application. Other Modules, such as the SmartPhoto and ButtonBar Module can be used to enhance the presentation of your application or web pages. There are just 8 types of connectable Module See Appendix 1.

You can use each type of Module any number of times. Each time you create a Module of a certain type, you configure or customize each one as required, and a new icon appear in your Module Store to represent the new Module you just created.

### DATA MANAGER

---

You can't buy all the Module individually. In order to create useful web applications you usually need a basic set of 6 Module types. We've bundled these together to form the **Data Manager** collection of Modules, that consists of the Form, Database, Database Reader, database Writer, Reporter and Mailer Modules.

### MODULE STORE

---

All your customized Modules are stored in your private **Module Store**. This is the center of the Xara Module development system. Each Module has its own icon in the Module Store. See below for mode details.

### MODULE EDITOR

---

You can customize the look and behaviour of any Module by clicking its icon in your Module Store. This takes you to the **Module Editor** for that Module. By clicking the **save & publish** button in the Module Editor the updated Module is saved back to your Module Store.

## HTML SNIPPET

---

Each customized Module has an **HTML Snippet** associated with it. You place this Snippet on your web page where you want the module to appear.

This HTML Snippet is created when you create the Module. It doesn't change if you edit the Module, so you don't need to edit your web page no matter what changes you make to the Module.

You can easily find the HTML Snippet for any Module by clicking the Module's name in the Module Store.

## FRONTPAGE ADD-IN DREAMWEAVER EXTENSION

---

If you use Macromedia Dreamweaver or Microsoft FrontPage, you can download a free Extension (Macromedia) or Add-in (FrontPage) which makes it easier to access Modules and your Module Store from within these programs. This saves you having to copy and paste the HTML Snippet for any Modules and provides a direct way to view your Module Store.

## EXCEL ADD-IN

---

There is a free Microsoft Excel Add-in that you can download that provides an easy, direct way to synchronise with any Database Module. This provides a easy way to view and edit data from your online web database.

Public web page  
Private web page

**Public web pages** are pages you want anyone to see. An example is a page containing a 'join our mailing list' form.

**Private web pages** are pages that only you or the administrator want to see. For example, if the application provides web-based access to the data and online ways of updating your website or data, then typically this will be a password protected **private web page**.

## PROCESSES

---

An Application is typically made from several parts, which we call **Processes**. Each Process represents a separate action within the Application. For example, if your Application has a Database at its core, then one Process might be adding a new entry to the database. Another process might involve finding and displaying some data.

## INTRODUCTION TO XARA MODULES

In your Module Store each process is displayed as a left-to-right sequence of connected Modules. When the process is run (usually by a user pressing a button on form on your website) the Xara Module system executes the process taking data from a Module and passing it to any Modules to its right.

## The Module Store

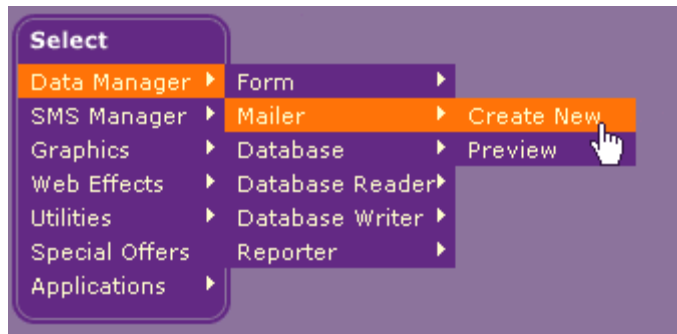
The Module Store is the heart of the Xara Module system. This is your private, password protected area where your Modules, your files, photos and other data are stored.



This shows an example Module Store containing a few connectable Modules (arrows indicate the direction of data flow from one to the next), two folders, a single ButtonBar Module and two photos. Options on the left allow you to create new Modules and operate on existing ones in your Module Store.

Just click on a folder to enter that folder. Each new Module is stored in the 'current folder', but can be copied and moved around using the tool options on the left.

To create a new Module select from the various Module categories, select the Module type you want to create and the 'create new' sub-menu.



## Products

The 40 or so different types of Modules are grouped into different product categories. For example there is a graphics category containing many graphics Module types such as the **Heading Module**, **Button Bar Module**, etc. As with all Modules you can create a new Module, customize its appearance using the online **Module Editor**, and save this back to your **Module Store**. You are provided an **HTML Snippet** that when inserted into your website will make the Module appear on your web page.

Some Modules are provided free of charge, such as the Visitor Counter Module, RealTracker and Banner Manager Module. Some Modules you can purchase individually, but most are available as part of a collection. For instance the 'Web Graphics & Effects' collection consists of more than 20 different Module types purchasable as one group. Once you have purchased any module you can create as many examples of each Module type as you desire.

The various connectable Module are available in 3 different Module collections;

The **Form & Mailer**, as the name suggests consists of just the Form and Mailer Modules and is designed for those that just want to collect data on a form on their website and have this emailed.

The **Data Manager** collection consists of the core connectable Modules, Form, Mailer, Database, Database Writer, Database Reader, Reporter.

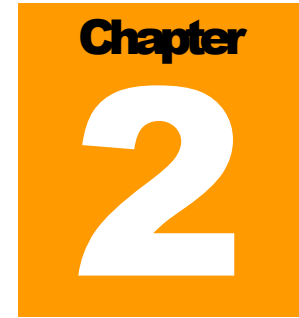
Click the **SMS Manager** consists of the 2 SMS Texting Modules required in order to implement any texting functionality to or from your phone.

Click the **Buy Now** button in your Module Store to see the full product range. This includes not only all the Modules, but a number of pre-created web applications ranging from the simplest Feedback Form to the most complex examples such as the Membership Manager.

## **Usage Quotas**

When you purchase your Modules you are provided with a certain usage quota. For example the Database Module supports up to 2000 records for the basic monthly price (this is across any number of separate databases). The Mailer Module enables up to 2000 emails per month to be sent. Your basic Module Store space is 20Mbytes (this is mostly used by uploaded photos and other files). You can purchase additional quota 'Plus Packs' to support larger quotas. Click the Buy Now button in your Module Store to see the various quota plus-packs available.

The SMS Modules have a small per-use charge for sending or receiving each SMS text message. You can purchase additional message quota blocks as required.



## Xara Modules Architecture

*This chapter covers:*

- Dataflow Architecture
- Inputs & Outputs
- Fieldnames and Records
- Application and Processes
- Data Types
- XML Web Services

### Dataflow Architecture

The Xara Module architecture is unlike traditional computer languages. There are no specific controls for looping or branch control. If you're a programmer experienced in traditional programming languages, the architecture needs a little getting used to, but once you are used to it, it can be hugely more productive. If you're not an experienced programmer we have a system that's much approachable, has a substantially lower leaning curve and enables you to create online solutions previously only possibly with complex programming environments such as ASP, Perl, Cold Fusion or other traditional programming methods.

The Xara Modules architecture is primarily a data-flow architecture. When designing applications you should consider where you want the data to flow. When connecting Modules to create your solution, you 'wire up' the Modules so that the data flows from one to another in the required order.

## XARA MODULES ARCHITECTURE



In the simplest case when you click a submit button on a form data is sent out of the corresponding Form Module to the next Module in the chain, in the above case the Mailer Module.



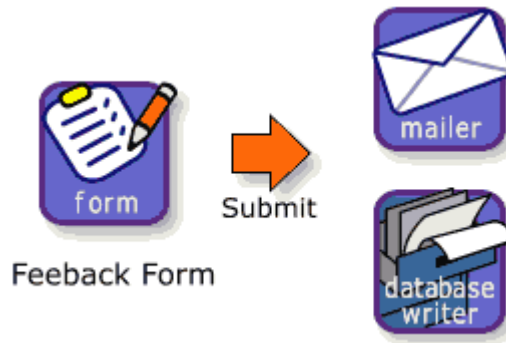
In this real-world example of retrieving some data from a database and displaying this on screen you can see the first stage involves entering some search criteria on a search form. When the 'search now' submit button is pressed this search criteria is passed to the Database Reader Module which retrieves the required data from the database and then passes this to the next stage, a Reporter Module which is used to display the results on screen.

## Inputs and Outputs

Modules can have inputs, outputs or both (depending on the type of Module). When your application is processing some data it sends the data from its output to the input of any other connected Module, or Modules. Diagrammatically this is always shown in the left to right data flow.

When you are customizing a Module, you connect its inputs to the outputs of another Module, and so the usual process is to create the Modules in a logical left-to-right order, in same order as the data flows. So in the above example the Form Module would be created first, followed by Database Reader Module, followed by the Reporter Module.

It's not uncommon to have the output of one Module connected to the inputs of two separate Modules, as in this example;



In this case when the submit button is clicked the form data is sent to in the input of both the Mailer Module (resulting the form data being emailed) and to a Database Writer Module resulting in the form data being saved in a database.

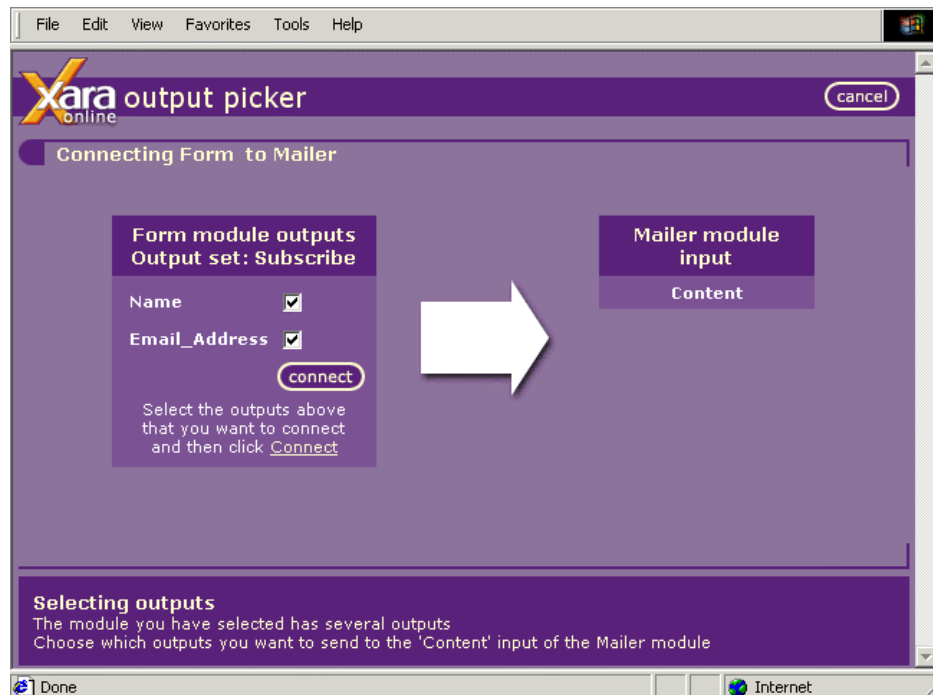
Full details of the inputs and outputs of each of the Modules are given in *Module reference Appendix*.

## Fieldnames and Records

The data passed between Modules consists of one or more records. Each record is made up of separate fields. The simplest example is a Form Module where the output is a single record consisting of as many fields as you have on your form.

The Database Reader can output multiple records if the search results in multiple matches. Most Modules that have inputs (Database Writer, Reporter, Mailer and Web2SMS) that are designed to cope with receiving multiple records. For instance the Mailer Module can be configured to send multiple, separate emails if it receives multiple (address) records, or it can be configured to place multiple (message) records into the content section of one email.

Each field that makes up the record has a unique fieldname that is used to identify that field or the data coming from that field. When you create a Form Module each cell is automatically given a fieldname (based on the title or description given to that field). When connecting a Module's outputs to another's inputs you are shown the fieldnames.



*When connecting the output of the Form to the input of a Mailer you can see the Form outputs two fields called 'Name' and 'Email\_Address'. Both are being connected to the Mailer input.*

It's highly recommended that you think of and use fieldnames as meaningful identifiers of the data throughout all processes that use this data. In other words if you are passing an email address field from Module to Module, you always call it something like 'Email\_Address' and not sometimes 'Email' and sometimes 'emailaddress'. This not only makes it much easier to understand what's going on but helps the system automatically connect the correct outputs to inputs where it can.

## Applications and Processes

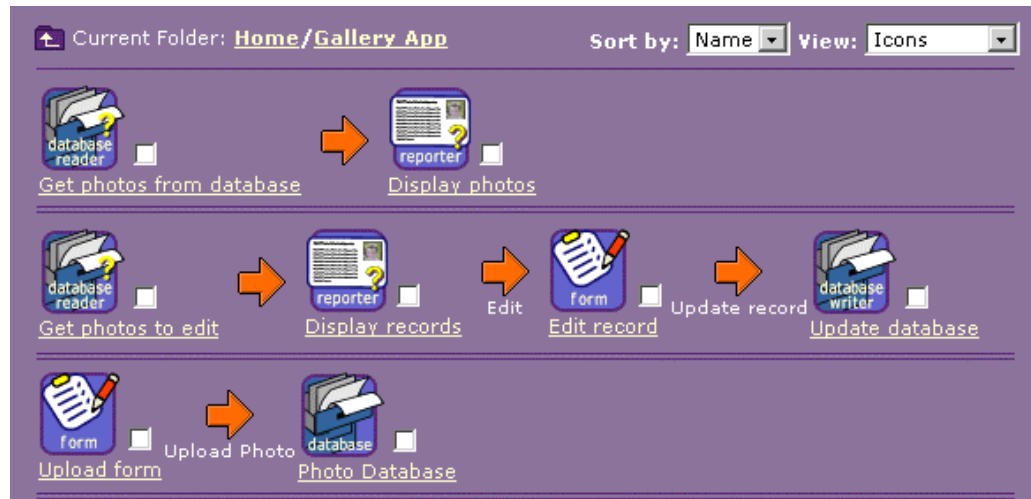
An **Application** is the overall system that you create by linking Modules together. An example of an Application might be a Mail List Manager. Part of this application might let people add themselves to your mailing list. Another part may let them update their details. Another part might involve sending a newsletter to everyone in your mailing list. An Application can be as simple or complex as you need it to be.

Each part of the application is called a separate **process**. Each Process represents a separate action within the Application. Each process is executed independently of the other processes.

In your Module Store each process is displayed as a single left-to-right sequence of connected Modules. When the process is run (usually by a user pressing a button on form

## XARA MODULES ARCHITECTURE

on your website) the Xara Module system executes the process taking data from a Module and passing it to any Modules to its right.



In this Module Store example you can see three separate processes. The top one simply retrieves photos from the database and displays these on screen.

The second process, the most complicated, represents the process of editing a record. It first gets all the records, then displays these on screen using the Reporter Module, which allows the user to select one to edit, which in turn is passed to a form where they can edit the details, and upon submitting that form the new data is sent back to the database.

The third process is a very simple one that allows new records to be added to the database, and consists of a form connected to the Database Writer.

## Data Types

There are different types of data that are passed from Module to Module. The most common is **simple text**. Most data entered into a form will be text and most useful data stored in your database will be text. Sometimes the text is a single line (such as a name or email address). Other times the text consists of multiple lines, such as a product description to the content of an email.

The database can also store other types of data such as **numbers, dates, HTML** and advanced types such as **Module Snippet** and **Yes/No** (boolean) types. In most cases you enter these as text and they are shown as 'text' when displayed. However you can perform certain actions on 'numbers' or dates that you cannot on 'text'. For example, on a form you can restrict input to a number or a date within a specified range.

When you create a Form and add new fields to the form you get the chance to select the data type required. When this is connected to the database for the first time it will

automatically configure the database structure to match these data types. You can also define and re-define if necessary the types directly from within the Database Module editor.

Below is listed all the data types supported.

---

### **SINGLE LINE TEXT**

---

The simplest and the most common type of data. E.g. a name

---

### **MULTI-LINE TEXT**

---

Used when you want to store one or more paragraphs of text. E.g. the content of a plain text email would be passed as multi-line text.

---

### **MULTI-LINE RICH TEXT**

---

A special version of the above this is basically HTML, so you can include text formatting and attributes such as bold, different fonts colors, etc. Usefully the Form Module allows a full rich-text editor that can be included on any form on your website and so it's easy to let users input rich text with formatting to be used on any Modules web application. If rich text is sent to the Mailer Module content this results in an HTML mail being sent. If rich text is sent to the Reporter Module then this is displayed exactly as entered.

---

### **NUMBER**

---

To be used where any numeric data need be passed, particularly where you might want to perform comparisons or searches for ranges etc. It is *not* recommended you use this format for things like zip-codes, phone numbers etc. You have the option of entering integer or floating point numbers.

---

### **DATE**

---

Used where you need to store or operate on dates.

---

### **SINGLE ITEM SELECTION**

---

An example might be choosing a day of the week where the user can select one of seven possible options from a drop-down list. Single item selections can be entered on a form using either drop-down list or a group of radio buttons. (When you design the Form you can select which type you require. The drop-down list is a more compact way displaying a larger list of items to select from than a radio button group).

## **MULTIPLE ITEM SELECTION**

---

For use where the user might want to select multiple options. These are always displayed as a set of check boxes on a form. The user sees all the available options and so check boxes are suitable where you have a limited range of options.

## **YES/NO**

---

Where you want to display just a single check box. The check box can either be selected (yes) or not (no). Equivalent to the traditional binary or boolean data type (but we like to keep the jargon down :-)

## **XML Web Services**

*For advanced users.*

If you're familiar with the concept of XML Web Services it sometimes helps to understand the system by realising that each Module is actually a separate XML Web Service. When you create a new Module you are in reality creating a new customized instance of that Web Service – the icon in your Module Store represent that instance of the Web Service. The Module editor controls what the web service will do and what data it expects on its inputs and what data it will provide on its outputs. When connecting Modules you are controlling the data that flows between each Web Service, and the order in which each Web Service is called by the Module system.

When a process is executed the system (we call it the Connection Layer) obtains all the data required in order to call the Module Web Service, or Modules if there is more than one vertically on a column, and then calls each Module passing the required parameters to the Web Service and collects the returned data from the Web Service, ready to pass to the next Module or Modules in the chain.

Typically a process is initiated when the user performs an action such as clicking on a submit button on a form. However it can also be executed when the system received an incoming SMS message on the SMS2Web Module (which is conceptually just like a user submitting a form). Certain Modules, such as the Database Reader can also be configured to automatically trigger a process when the data in the database has changed.

Xara Modules can interface with external third party XML Web Services where required (for instance the Excel add-in database synchronization tool communicates via a secure XML Web Services interface to the Module system). Please contact Xara Online support if you have specific requirements in this area.

## Web Pages and Redirects

Most web applications consists of multiple web pages. As the user interacts with the application, filling in forms and clicking on links they are taken from page to page as part of the process. When designing your application it's vital that you understand this and make a plan of the pages you require for each step of a process.

The basic thing to remember is that when user press a submit button on a Form they will be redirected to another web page. Similarly if they click a submit button or a 'select record' hyperlink on a Reporter Module that will also redirect the user to a new page. You need to decide what this page will have on it, and also to create the web page using your web authoring tool and know what the page URL will be.

When you create a Form Module, specifically when you insert a submit button on that form, you have to set the redirect URL.

The screenshot shows a dialog box titled "Cell content: Submit button". At the top right is a "Change Cell Type" button. Below the title bar, there are several fields and options:

- Button text:** A text input field containing "Search now". To its right are "font" and "Default:  ?" labels.
- Redirect to:** A dropdown menu showing "displayresults.htm" with a "?" icon to its right.
- Below the dropdown are three radio button options:
  - Current window, frame: [empty text field]
  - New Simple Window | Width: 640 | Height: 480
  - New Browser Window | Height: 480
- Color:** A label "Background" followed by a small square icon.
- Cell content:** Two small icons representing different cell content types.
- Horizontal:** A dropdown menu set to "Center align".
- Vertical:** A dropdown menu set to "Top align".

At the bottom of the dialog are three buttons: "OK", "Cancel", and "Help".

This shows the editor for the submit button of a Form Module. You can set the text on the button ('Search now' in the above example) and also the redirect URL. In this case, and we'd always recommend, that this is a relative URL, e.g. just 'displayresults.htm' as in this

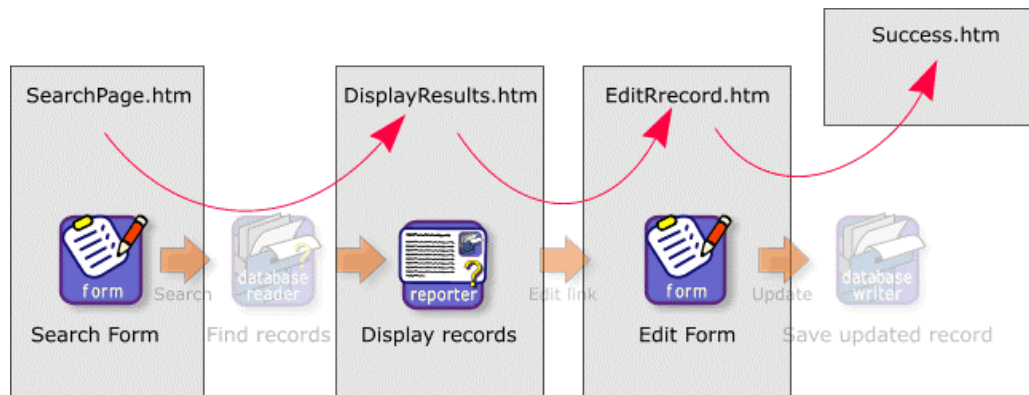
## WEB PAGES AND REDIRECTS

example. This means after they click the submit button they will be redirected to a page called 'displayresults.htm' that's stored in the same folder on you web server as the page you're coming from.

Typically this page would contain a Report Module which would show the search results. The Module system is smart and wont redirect to the new page until the search results are available so you don't have to worry about timing or the user seeing the page before the results are ready.

When putting a record selector onto a report (e.g. an edit link) this will also redirect to a new page when they user clicks the link. This has an almost identical editor as shown above where you can set the page URL to redirect to.

Lastly when you redirect you can control whether the redirect takes them to a new page in the current browser window (the usual case) or creates a new pop-up window instead. You can also control the size of that window.



This diagram illustrates the steps taken by the five Module 'edit a record' process described in more detail later.

The first page 'SearchPage.htm' contains the Form Module. When the user clicks the search button their browser is redirected to the 'DisplayResults.htm' page which contains the Reporter Module which shows all the found records. The user selects a record to edit by clicking the appropriate 'edit' hyperlink on the report. This will then redirect them to the next page 'EditRecord.htm' which contains the edit Form Module where they can now edit the record details. Upon clicking the update button they would be redirected to whatever page you desire. In this case they are taken to a 'Success.htm' page that might display a confirmation that the record has been updated. This last page doesn't have any Modules on it and would just be a landing page created for the purpose of telling the user the process has completed.

## How to Design your Application

*This chapter covers:*

- Which Modules to use?
- The Database, Reader and Writer Modules
- Database synchronisation
- Suggested steps to design your application
  - 1) Sketch your process schematic
  - 2) Sketch your Form design
  - 3) Create your Modules
  - 4) Test the Modules and process
  - 5) Integrate the Modules into your website

### Which modules?

#### FORM MODULE

---



It should be self evident when to use the Form Module. Anywhere where you require the user to input some data you will require a Form and hence a Form Module.

Forms always have a submit button which when clicked submits the form data to the Module system which will pass this data to the next connected Module/s.

The Form Module editor lets you design just about any complexity of Form with a wide range of different field types from simple plain text input fields to drop-down lists, radio buttons and even a powerful WYSIWYG Rich text editor (HTML editor).

Perhaps less obvious is that when the user edits some data, this is also done via a Form, but in this case the form is not usually blank but is pre-populated with the data you want to the user to edit. This is done by passing the required data to the Form Module inputs before the Form is displayed. (More details later).



## MAILER MODULE

---

Wherever you require to send one or more emails you will need a Mailer Module.

The simplest application consist of just these two Modules, a Form Module connected to a Mailer Module. An example walk-through of this process is described in more detail in a later chapter.



## WEB2SMS AND SMS2WEB MODULES

---

These two Modules are used when you want to send or receive SMS text messages to or from a mobile (cell) phone.

The Web2SMS acts just like the Mailer Module, and accepts inputs which are then immediately sent as a SMS to a phone. The SMS2Web accepts incoming text messages and outputs the message on Module output (in this context it acts rather like a Form module and can be used to trigger a process). Refer to the Appendix for more details.



## REPORTER MODULE

---

This could have equally be called the Display Module. It's used in almost all situations where you want to display data on a web page. (It's called The Reporter because when displaying data from a database it's been traditional to call this a Database Report).

Not surprisingly this Module is used to display any data from a database, but it's a lot more powerful than this. It can be used to format just about any data into tabular form for display on screen, or send this formatted data (as HTML) to another Module such as the Mailer Module.

One key use of this Module is to let users select one or more items from a list for further processing. For example is it can display a list of items with an hyperlink against each item that when clicked send that selected item to the next Module in the chain. You'd use this for selecting records to edit, or email etc. Alternatively it can display a list of items with check boxes next to each item (Hotmail like) where you can check multiple items to select and process.

## Database Module Database Reader Module Database Writer Module

These three Modules form the heart of most applications.



## DATABASE MODULE

---

If you need to store any information in your application you will require a Database Module. You can have as many Database Modules as you require. For example you may need to store a list of email addresses, and separately a list of products. This would require two Database Modules.

**Think of the Database Module as primarily the data store. This is where the actual data of your database is stored.** You can actually view and edit (and upload and download) the data stored in your database by using the Database Module editor. You'd create a separate Database Module for each separate set of data you want to store. For example a name and address list would use one Database Module. A separate list of products would use another Database.

There are two closely associated Modules; the Database Reader and Database Writer Module. It's important to understand that Database Module is primarily used as a data store and the other two Modules are used to save data to the database (Writer) or to search and read data from the database (Reader). The Database Module, for example, has no connected outputs – it never outputs data, this is the job of the Database Reader.

When you first connect a Form to a new, blank Database Module this defines the database structure, namely the fieldnames and data types.



## DATABASE WRITER

---

If you want to save or update any data in your database you will require a Database Writer Module. The inputs to this Module, rather obviously, are the data you want to store. When you create the Module in the first place you specify which actual Database Module you want the data stored in.

It's quite common that you have more than one Database Writer, all writing to the same database. For example you may have a process where you add new records, and separately have another process that lets end-users edit records. This Module has no outputs, only inputs.

There is a shortcut that avoids using this Module in some situations. The Database Module can double up as a Database Writer in the simple case where you simply want to add new records. If you want to update an existing record then you must use the Database Writer Module. If you need to add records in more than one place (you might let end-users add records in one place and separately let the administrator edit records from other password protected pages) then you might use the plain Database Module in the first case where each new record is simply added to the database. You'd use the Database Writer in any secondary situation where you want to save or update records.



## DATABASE READER

---

As the name suggests you will need a Database Reader Module in any situation you require to search or retrieve any data from your database. When you first create this Module it asks you which database you want to read from, so from that point on that Reader will be directly associated with that specific database.

This Module can perform advanced searches of the data and only retrieve the records that match your search criteria.

This Module has both inputs (the search criteria) and outputs (the records that match the search criteria). It's also possible to configure this Module with preset search criteria so that it has no inputs. E.g. you might configure it to read all records in your database (a pretty common operation) say for the case where you want to send an email shot to all email addresses in your email list database.

This is one of the most powerful Modules and is used extensively in just about all applications. It's not uncommon to have multiple separate Reader Modules reading from the same database.

(This Module was previously called the Database Query Module)

## Database Synchronization

Xara provide a free Microsoft Excel add-in that enables you to keep a copy of your database on your desktop and to use Excel to view edit or update that data. You can synchronize this with any Database Module so that your local copy and the online Database are always the same (the synchronization process copies any updated records to the online database and any updated online records back to your local copy).

This is a great way to download updates or new records that might have been added to your online database, and also a very quick and convenient way to edit records in your online database.

## Suggested steps to design your application

While it's certainly possible, especially if you're familiar with the Modules system, to build your application as you go with no planning, we highly recommend spending some time on planning. The basic steps are described below:

### Step 1: Sketch your schematic

Once you understand which Modules to use you should be able to put together a rough schematic illustrating the Modules required for each process of your application.

## HOW TO DESIGN YOUR APPLICATION

Remember each process represents a single user-action. Here are some examples of common processes.

### Emailing Form contents



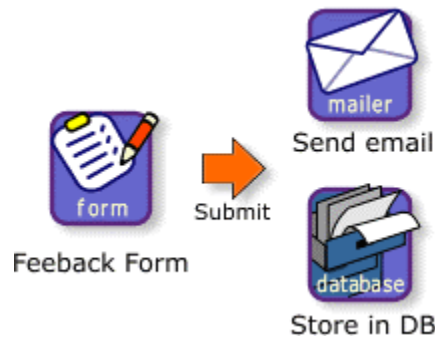
As simple as it gets. The output of the form, such as a feedback form, is connected to the input of the Mailer Module, which is configured to send the contents of the form to a given email address. When creating the Mailer Module you'd configure the email address the address the data is sent to and would connect the Form output to the content section of the email.

### Storing form data in a database



If you want to save the form data in a database simply connect the output of the Form Module to the input of a Database Module. You'd create the form first with all the required fields and then create the Database module and connect it to the Form. The first time you do this the database is constructed to contain the required fields and data types. Connecting directly to the Database Module is a shortcut only where you want to add new records always. Normally you'd connect to a Database Writer Module to save data to a Database.

### Storing data and sending an email



This represents the case where you want the data from the form sent in an email and at the same time stored in a database (perhaps for future reference for searching). When you create and configure the Modules you define what parts of the incoming data goes where. Usually you'd store all the form contents in the database, but you can define only to store

## HOW TO DESIGN YOUR APPLICATION

certain fields. Again we're using a Database Module, rather than a DB Writer since we just want to add each new form submission as a new record.

### Search your database



In this simplest case you can create a search form (which would have fields you want to search on) and connect the output of this to a Database Reader. When you click the search button on the form the search criteria fields are sent to the Database Reader which will search the appropriate database and return the matching records. If the Reader Module has no further Modules connected to its output (see below) you can make this Reader Module display the results on screen (in a simple table format).

### Search your database and display formatted results



Where you want your search results presented in a specific formatted way, then you'd connect the output of the Reader Module to a Reporter Module and configure this to display the records as you require.

When the user clicks the search button on the search form, this causes the search criteria to be sent to the Reader which recovers the matching records and sends them to the Reporter to format and display the results on screen. You might be wondering so how exactly do you see the results. The answer is that the HTML snippet associated with the Reporter is placed on your 'search results' page. When the user submits the Form you redirect the user from the page with the search form on to the page with the Report on to see the search results. (see a later chapter on web pages and redirects to learn more).

### Automatic updating of a database listing



The Database Reader Module can be configured to be triggered automatically each time the data in the corresponding database is altered. This means a simple process of

## HOW TO DESIGN YOUR APPLICATION

automatically updating a displayed database listing can be as simple as the two Modules shown above.



This represents the entire process of editing a record and is one of the more complex processes you're likely to come across.

The first three Modules are exactly the same as above (search form, find and display results). However in this case the Reporter would be configured to display 'edit' links against each record which when clicked will send the output of the selected record to the edit form where they can edit the record. When they click the update button on this final form, the contents of the record are written back to the database using the Database Writer.

This process would usually involve three page redirects. Firstly when they submit the search form they would be redirected to the page with the Report Module that displays the found records, when they select a record to edit they would be redirected to the page with the edit form on. Finally when they click the final update they would be redirected to whatever page you desired as the next stage (perhaps just back to the first page of the process).

### Update your website



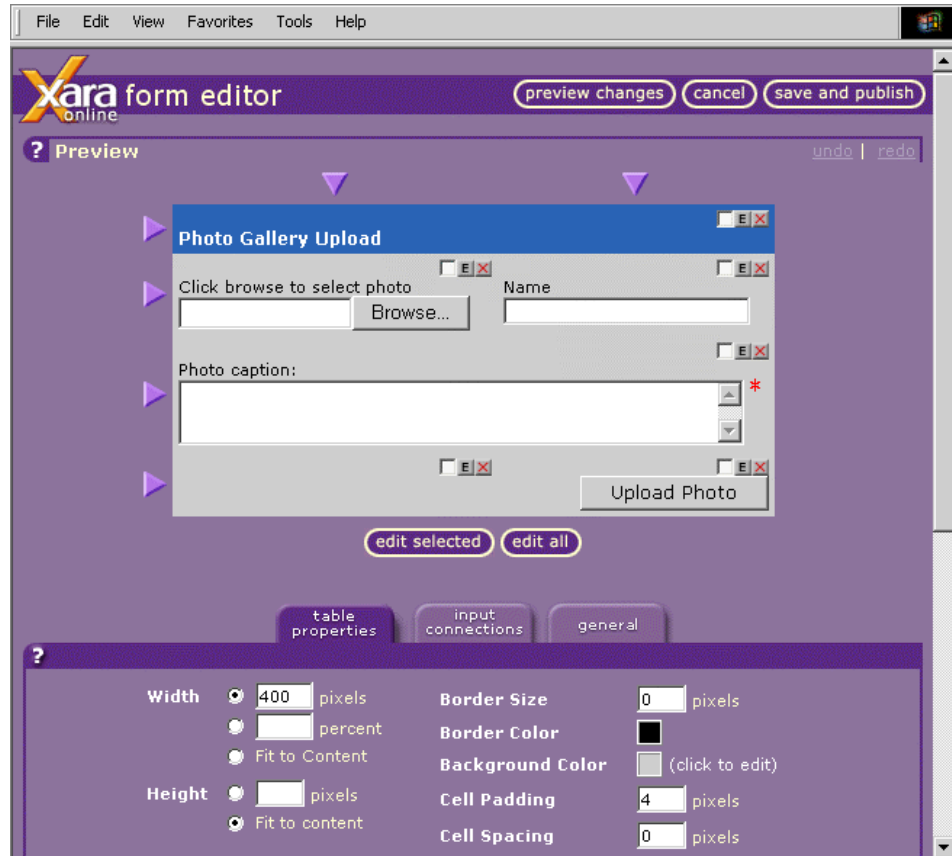
This is a rather unusual, but very simple application that allows sections of a website to be updated remotely from your phone. The SMS2Web Module received text from your phone and is connected directly to a Reporter Module which is placed on your website. Instead of the SMS2Web module you could use a Form Module which would provide a very simple way of updating sections of your website from a form on any web page.

## Step2: Sketch your Forms

You can save a great deal of time by drawing your proposed form on paper first. The Form designer (i.e. the Form Module editor) is very powerful but changing the layout of

## HOW TO DESIGN YOUR APPLICATION

an existing form can be quite time consuming. Far better to get a good idea of what you want your form to look like first.



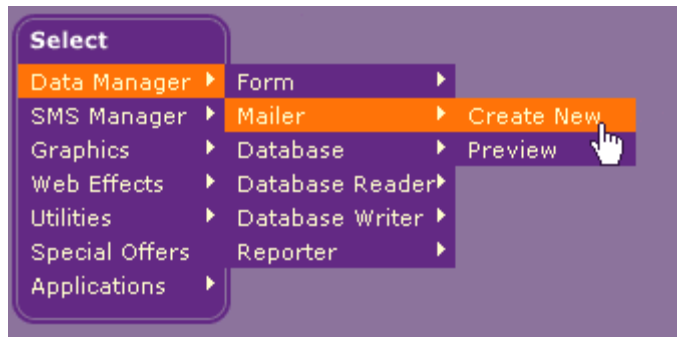
The Form Module editor allows you to design forms based on a table grid. You can insert rows and columns and insert any one of the 16 different types of fields into each cell. You have full control over form size, color, fonts and more.

### Step 3: Create your Modules

We recommend creating the Modules for one process at a time, in the same left-to-right order in which data flows from Module to Module.

Create each new Module using the Module Store menu as shown.

## HOW TO DESIGN YOUR APPLICATION



You will be taken to the Module editor where you configure the operation and appearance of the Module. From the editor select the 'save & publish' option to create the Module where you will be given an HTML snippet to add this module to your web page. (If using the FrontPage or Dreamweaver add-ons the Module is inserted directly into your web page). For each new Module you create a new icon is added to your Module Store.

Some Modules do not have any useful component that appears on the page – that is they operate in the background. E.g. the Database Modules, the Mailer and SMS Modules. When these Modules are first created you are still given a HTML snippet and a simple status panel is displayed. You can safely delete this from your web page if desired as the Module will continue to operate normally on the server.

Other Modules such as the Form and Reporter Module have obvious visual component that forms the user-interface of your application.

Refer to the online help and to the Modules Reference Appendix of this guide for more detailed information about the operation of each Module.

## Step 4: Test the Modules and Processes



You can preview any Module directly from the Module Store by clicking the Module

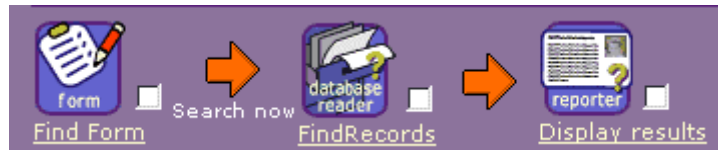
## HOW TO DESIGN YOUR APPLICATION

name hyperlink under each Module. This pops-up a window showing the Module exactly as it will appear on your web page. It also displays the HTML snippet associated with the Module.

You can also use the preview windows to test the functioning of your Modules and whole processes.

This is particularly useful for the Form Module where you can actually enter test data and click the submit button to test the process. Assuming the Form Module is connected to other Modules this will start the process.

You can then click the preview link of further Module down the chain to view resulting actions. For example if you had this process:



By clicking on the Form Module preview link and entering some test data it will start the process. Once the form has been submitted you may get a redirect error in the window since the preview window cannot redirect to the correct web page (see below). Just ignore the error and close the preview window. The form data will have been sent to the Database Reader Module which in turn will have sent the results to the Reporter Module. You can now click the preview link of the Reporter Module and see the resulting data. (The Reporter Module cannot show a correct preview until it has been sent some data to actually preview).

You can then edit the Form or Reporter as much as you like to tweak the presentation of either until you are sure you've got the process working exactly as you require.

### Web Page redirect testing

The one aspect you can't test using the preview windows from the Module Store is that the page redirects are working correctly. This can only be tested by embedding the Modules into real web pages with the correct page names so the redirects will work correctly.

### Testing database writing

If you have a process that writes data to a database, the easiest way to test this has worked correctly is to go into the Database module editor (i.e. just click on the Database module icon) from where you can review, and if necessary add or edit any record.

### Testing Reporter Modules

The Reporter Module takes data sent to it and formats it into a table. In order accurately preview what the Report will look like (either in the Module editor itself or the Module preview link) it's necessary to send some real data to the Module.

## **Step 5: Integrate the Module into your website**

If you are using the FrontPage or Dreamweaver add-ins the Modules will already be inserted into your website when you first created the Modules. In this case it may be necessary to move the Modules to the correct web pages (just copy and paste as usual).

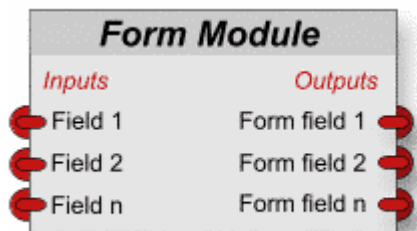
In all other cases you can simply copy and paste the HTML snippets (from the preview window) for the Modules you require to appear on your website into your web authoring tool.

You can now test the page redirects work correctly by previewing or publishing your web pages. Note that if you are using relative page URLs for the redirects in the Form and Reporter Modules that your web pages must be stored in the correct relative position – typically this means the HTML pages are stored in the same folder so you can use a simple relative page name.

You should now have a fully working process that can be deployed to your live servers.

## Reference

# Form

**PURPOSE**

A customizable form to appear on your website. Accepts input from website users and outputs the contents of the form to any connected Modules.

It can be designed to have any number of fields and to accept data of just about any type from customers or website visitors. You can customize just about all aspects of the appearance of the form which is designed on a grid basis with each cell of

the grid containing any one of the following field types:

- Text
- Single line text input
- Multi-line text input
- Multi-line rich text input (full HTML editor)
- Number input
- Date input
- Drop-down list (single selection item)
- Checkboxes (multi-selection items)
- Single checkbox
- Password input (shows \*\*\*\*)
- Module HTML Snippet
- File upload
- HTML file upload
- Submit button

The font type and size used for any fields (the field title and the field itself) is exactly controllable to a point or pixel resolution.

**WHAT APPEARS ON YOUR WEB PAGE**

The form.

## **INPUTS**

---

If you want the form to appear blank then there is no need to use any inputs. This is the most common case.

If you want the form to be pre-populated with content when it's displayed e.g. if you're editing a record, then you can connect each of the form fields to an output of another Module. Typically this would be a Database reader Module.

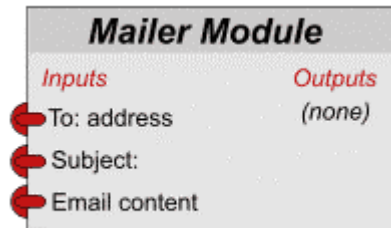
## **OUTPUTS**

---

When the visitor clicks the *Submit* button, the contents of the form are sent as the output of the Form Module. The output consists of one record made up from various fields – each field corresponding to a field on the form.

# Reference Mailer

## PURPOSE



Sends an email for each record it receives on its inputs.

The Mailer can be configured to use preset values for each of the To:, Subject, and Content fields. Alternatively these can be connected to the output of another Modules.

So for example if you want to configure the email to always go to a set address just set the To address (and probably Subject field) and then connect the Content to the input.

If this Module receives multiple records To: addresses on its input (e.g. as a result of a Database Reader output) then it will send multiple email, one for each record. In this case it's expected the To: address is passed as an input so it knows which address to send each email.

The Module supports mail merge allowing personalized email. It also supports HTML or plain text emails. If sending HTML email this can be pre-configured or passed in as an input, in which case the HTML can come from an uploaded HTML file, HTML stored in the database or from the output of a HTML field on a form.

The Mailer support automatic remove requests (a reply to any email sent from this Module with 'remove' or 'unsubscribe' in the subject line) and will mark the corresponding database entry with a \*remove appended to the email address field. All other replies will be forwarded to the account holder's email address.

The Module will automatically retry sending any bounced email 3 times (in the case where the recipients email server is down temporarily).

## WHAT APPEARS ON YOUR WEB PAGE

Like the Database Module this operates mostly in the background and displays a simple status box on your web page displaying the number of emails sent. This status information can be safely deleted from your web page and the Mailer Module remains safe and operational. The only way to delete this Module is directly from your Module Store.

## **INPUTS**

---

This module has three inputs:

**To:** address. Expects an email address, or one email address per record if there is more than one record. It does not expect or allow comma-separated addresses.

**Subject:** Expects a single simple string that is the subject line. Usually this would be preset in the Module to be a fixed subject line such as 'Thank you for subscribing' rather than being connected as an input.

**Content:** This can be configured with preset email content text or connected as an input. If this Module is receiving bulk record data on its inputs, then it sends the same email content to each of the email addresses in the bulk data (assuming the *To:* field is connected to the right email address output)..

## **OUTPUTS**

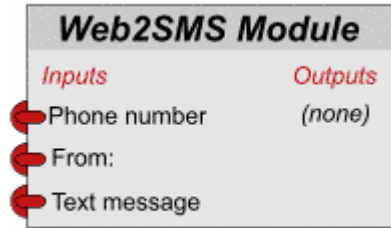
---

There are no connected outputs from this Module (it sends emails of course, but they are not 'connected outputs').

# Reference

# Web2SMS

## PURPOSE



Sends a SMS text message for each record it receives on its inputs.

The Web2SMS Module can be configured to use preset values for each of the Phone, From and Text Message fields. Alternatively these can be connected to the output of another Module.

So for example if you want to configure the Module to always send a message to the same phone, preset the phone number fields. SMS messages are limited to 160 characters only – if you send more text to the text message field it will be truncated.

If this Module receives multiple records on the phone number inputs (e.g. as a result of a Database search) it will send a separate text message for each record.

There is a small per-message charge for each message sent. You can purchase a messages quota using the Buy button in your Module Store. At the time of writing the costs starts at 12c (about 8p) per message decreasing depending on the volume purchased.

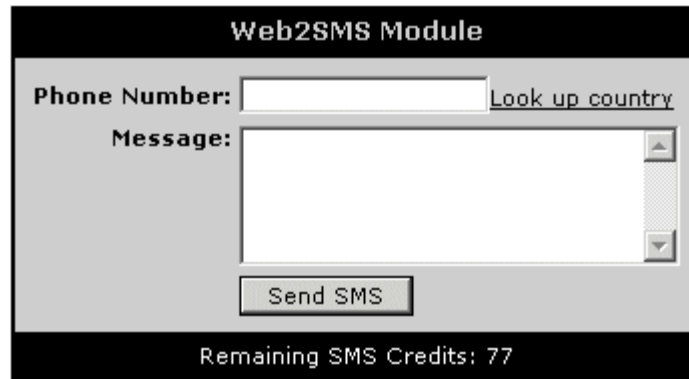
Messages are usually received within a minute or two of being sent, often within a few seconds, but this can't be guaranteed since it's dependent on the traffic on the individual phone company's SMS message gateway.

The Web2SMS Module can send messages to most countries in the world for a fixed price, using more than 500 different Network Operator Carriers.

## WHAT APPEARS ON YOUR WEB PAGE

When configuring this Module you can control what you want to appear on your web page. When working within a connected web application you probably don't want anything to appear on your page, in which case you can delete the Module form your web page. It will continue to work in the background.

Alternatively this Module can display a form like this;



Which lets you send SMS messages directly from a form on your website. This is useful if you just want a quick and easy way to send SMS messages from your desktop or website without using any other Modules. (be warned though that each message costs you money and so this is not something you'd want to put on a public website).

## INPUTS

---

This module has three inputs:

**Phone number: Important:** The phone number field must be in standard international format preceded by the country code. E.g. to send a text to a UK mobile number such as 07831 123 456 you would set the number to be 447831123456. i.e. use the UK country code (44) and omit the leading zero. To send to a US cell phone use the country code 1 but keep any leading 1 codes making sure the number is 11 digits.

**From:** This field can be set to a string or number. This is shown on the phone as the from text. They can only reply to the message if this is set to a phone.

**SMS Text message:** Expects a simple short text string (plain text not HTML). This should be no more than 160 characters.

## OUTPUTS

---

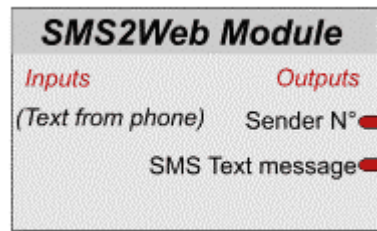
There are no connected outputs from this Module (it sends the SMS text message, but this is not a 'connected output').

# Reference

# SMS2Web

## PURPOSE

---



This is an input Module that accepts a short text message from a cell or mobile phone and send the message as an output to any connected Modules.

Messages are sent to a fixed phone number (to the number +44 7740 706 780) with the first word of the message set to a specific keyword.

This keyword is used to identify the correct Modules recipient. The remainder of the text message after the keyword is sent as an output of this Module.

When the Module is first created you are assigned a random 4 letter keyword that must be used as the first word in any test message. For a fee you can purchase specific keywords such as 'taxi'.

## WHAT APPEARS ON YOUR WEB PAGE

---

This Module operates in the background and so displays a simple status message on your web page. This Module can safely be deleted from your web page and it will continue to operate in the background correctly. Use the Module Store Delete option to really delete the Module.

## INPUTS

---

This modules has no connected inputs. It receives its input from an over-the-air phone text message to the above phone number with an associated keyword.

## OUTPUTS

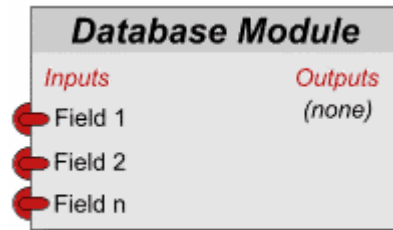
---

Upon receipt of an incoming text message this Module will output two fields, the sender's phone number and the text of the SMS text message.

# Reference

# Database

## PURPOSE



Stores records in an online database. You can have one or more separate database Modules to store different data.

It accepts data on it's input and adds these to the end of the database. (Use the Database Writer Module to replace existing records rather than append new ones).

This Module is where the actual data is stored and by clicking on the Module icon in your Module Store you can view, edit and add new records to your database.

Initially your Database modules are limited to storing 2000 records (across all your Database Modules), although you can purchase an additional 'plus pack' increasing your storage quota as required.

### Data synchronization

You can download a free add-in for Microsoft Excel that enables you to download and upload data to any Database Module directly from your desktop. This will synchronize Excel to the online Database and so is an easy way to keep a copy of your online data, to read and view all changes that have occurred to your online data, and to edit your database from your desktop.

To find out more and download the add-in click on the preview link of a Database Module.

## WHAT APPEARS ON YOUR WEB PAGE

This Module operates in the background and so displays a simple status message on your web page, showing the number of records used. This Module can safely be deleted from your web page and it will continue to operate in the background on the servers correctly. Use the Module Store Delete option to really delete the Module – but beware this will delete all data stored in this database.

## INPUTS

Accepts one or more records as output by other Modules, most typically the Form Module. Each new record received is added as a new record to the database. The database

## **APPENDIX: WEB 2 SMS MODULE**

can be configured to have up to 50 fields per record and each field can be connected to a separate input. The usual case however is that all fields are connected to corresponding outputs on a Form Module. When you initially create a database and connect it to a Form, it sets up the fields and all the connections automatically.

### **OUTPUTS**

---

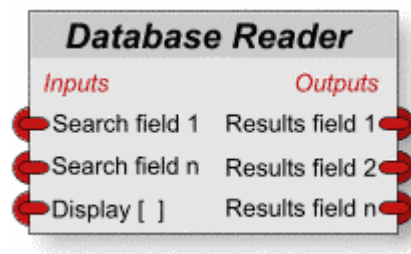
There are no outputs on this Module. (If you want to retrieve data from a Database Module use the Database Reader Module.).

## Reference

# Database Reader

## PURPOSE

---



This Module is used to search a database and retrieve one or more matching records and send these as outputs.

Each Reader Module is associated with one Database Module. Each Reader Module can perform a specific preset search, e.g. all those records added after a certain date, or it can be configured that the search parameters are

determined by the inputs to the Module.

## WHAT APPEARS ON YOUR WEB PAGE

---

This Module operates in two ways. You can configure it so that the Module itself has a basic search form, or alternatively you can configure it to use inputs to control the search.

## INPUTS

---

Alternatively you can control the search using a Form Module and connecting the form outputs to the query inputs. One input 'Display' determines whether the results of the search are displayed on screen or sent to the outputs.

Note: this Module retrieves the data from the database that is specified when the Module is first created, directly and not via connected inputs. It's best to view this Module as a database output Module.

## OUTPUTS

---

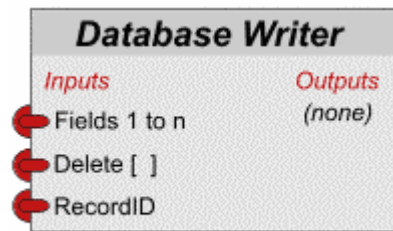
This Module outputs one or more records as determined by the search parameters. The fields contained in each record are determined by selecting the appropriate output checkboxes when this Module is configured.

# Reference

# Database Writer

## PURPOSE

---



This Module is used to send or store data in a separate Database Module.

When this Module is first created you specify a corresponding Database Module where the data will actually be stored.

If you have a simple application that just adds new records to a Database, it's easier and quicker to use the Database Module. However if you require to edit existing records (i.e. overwrite and existing record) or to delete records or require different processes to write to a common database then you will require a Database Writer.

## WHAT APPEARS ON YOUR WEB PAGE

---

Like the Database Module this operates in the background and just places a simple status message on your website. You can safely delete this from your web page and the Module will continue to operate correctly on the servers in the background.

## INPUTS

---

Just like the associated Database Module this expects to be connected to the output of another Module, and it will store this data in the Database Module associated with this writer.

In addition this has two additional input fields. The first is a RecordID field which is used to specify a record to replace. If this is not connected or blank the incoming data will be appended as a new record, in the same way the Database Module works. If however it's connected to a RecordID output of another Module the incoming record will replace the existing record. The RecordID is always output as one field from the Database Reader, and so if this field is passed through to the writer it can replace the required fields.

## **APPENDIX: WEB 2 SMS MODULE**

The second special input is the 'delete' field which expects a yes/no (i.e. a checkbox) connection. If this is set to 'yes' then the record specified by the record ID will be deleted entirely from the database. Use with caution.

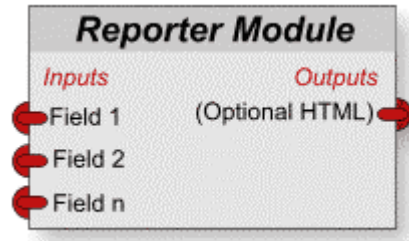
### **OUTPUTS**

---

There are no outputs from this Module. It simply stores incoming data in the associated database.

## Reference

# Reporter

**PURPOSE**

This Module is used to display data on screen in a formatted manner.

It accepts incoming data and arranges this in a repeating table of columns and rows. You can configure the presentation – which fields are displayed in which columns and the fonts, colors, text sizes etc.

Each new record is usually repeated below the previous.

You can automatically paginate the display so it puts a 'search engine like' page control at the bottom of the screen (e.g. [Previous](#) [1](#) [2](#) [3](#) [4](#) [Next](#) )

This Module can also be used as a record selector. Associated with each record you can add one or more hyperlinks which when selected will cause this Module to output the selected record to any attached Modules. This is called single record selection and is used where you require the user to select just a single record for further processing.

If you require the ability to select multiple records you can add a check box alongside each displayed record, and place a submit button on the Report. When they click the submit button all the selected records on this page will be sent as output of the Module.

**WHAT APPEARS ON YOUR WEB PAGE**

The Report. When you configure this Module you determine whether the Report is a 'public' report which means that all visitors to your website can see the report. Alternatively you can set it to be a private report in which case the Report is only available by the 'current' individual visitor who is operating this application.

The equivalent of a search engine. When you perform a search only you see the results – all other visitors to the search engine do not see your search results.

**INPUTS**

This expects one or more records to be sent as inputs. If there is more than one the Report repeats as many rows as required in order to display all the records (and paginates it is required). If there are more than 1000 records being input this will not display anything

and instead can ask the user to refine their search. (you can define the limit and the message)

## **OUTPUTS**

---

Under most circumstances this Module has no outputs, it just displays the Report on your web page where you inserted the Module. However you can configure it so that instead of displaying the Report it can send it as an HTML output of this Module. This is useful, for example, where you want to send a formatted Report in an email. In this case you'd connect the output to the Content section of a Mailer Module.