



# Deploying Sparks

May 2007

## **Deploying Sparks – A step-by-step guide**

This document outlines a simple process that can be used to get authors up and running on Firefly Web Server Edition.

### **Publishing your Spark**

1. Open *Firefly Designer* (if it's not already open)
2. Open the *Spark Map* you wish to publish – or create one as required (please refer to ***Creating your first Sparks*** in the appendix for information on this). Click on the **File** and **Edit settings...** menu items. You may wish to complete these details as they will help you identify the Spark when it is published onto the server.
3. Click on the **Options** and **Settings** menu items to check your author account login details. If you're not sure about your login details, please refer to ***Managing authors in a production environment*** in the appendix. You don't need to do this every time you publish, just the first time as outlined below:
  - a. *Web service address*: This is the address of your web services page on the server. This normally looks something like:  
<http://yourserver:8080/sparkstudio2007/services/SparkSoap>
  - b. *Username*: Your author account username (the username you use to log into Web Server Edition)
  - c. *Password*: Your author account password (the password you use to log into Web Server Edition). This will be blank – it does not indicate that a password has not been set – rather that the software cannot retrieve the original password once you have set it (it's encrypted)
4. Click **OK** to close the settings dialog
5. Click on the **Publish** and **Send** menu items to start the publishing process
6. If the settings are correct (as specified in step 3), you will see the progress of the Spark as it is sent over to the server. Note: If you need to go through a proxy server when publishing, please contact us for the relevant configuration information.
7. Click on the **OK** button once the submission is complete
8. Open your web browser and navigate to your installation of Web Server Edition (this will be something like <http://yourserver:8080/sparkstudio2007/admin>)
9. Enter the same login details you used in step 3b and 3c and click on the **Login** button
10. In the navigation menu down the left side of the page, click on the **Sparks** link
11. In the table, you will see your Spark with a *Registration type* of 'Not assigned'. This means the Spark is not yet available for use.
12. Click on the **Update** link beside the Spark
13. In the displayed form, change the *Registration type* to 'Public'.

14. Complete the details as required. For more information on these settings, please refer to [http://www.sparkmapping.com/sparkmap/manual/organise\\_sparks.jsp](http://www.sparkmapping.com/sparkmap/manual/organise_sparks.jsp)
15. Click on the **Update** button to save the settings. You will be returned to the page with the list of Sparks.
16. In the top right corner of the Sparks table, you will see a drop-down menu labelled *Template*. Select a web design template from the list – this will apply a look and feel to the Spark. For more information on Templates, please refer to, ***Setting up web design templates*** in the appendix.
17. Click on the **Run** link beside the Spark to open the Spark in a new window. If you have started the Spark before and did not close the window, it will appear in the existing browser window (you may need to find the relevant window in your task bar).
18. You can now navigate through your Spark as it will appear to the end user.
19. If you need to make any changes to the Spark, please repeat steps 5 to 17. This can be done as many times as required until you are happy with your Spark.
20. Your Spark can be accessed via a direct hyperlink or alternatively, you can give users access to your *User desktop*. For more information on the User desktop, please refer to [http://www.sparkmapping.com/sparkmap/manual/the\\_user\\_desktop.jsp](http://www.sparkmapping.com/sparkmap/manual/the_user_desktop.jsp).
21. On the Sparks page, you can generate the relevant hyperlinks by clicking on **Link page** link beside the Spark. This will generate an example HTML file containing the hyperlinks. For more information on what to do with the hyperlink, please refer to, ***Techniques for linking to your Spark*** in the appendix.
22. Once you have the hyperlinks added to your own web pages, you're all set to go!

## **APPENDIX**

### **Creating your first Sparks**

We have plenty of training and reference material available at our [www.sparkmapping.com](http://www.sparkmapping.com) website. We would recommend going through the Firestarter tutorials to help you get to grips with the authoring process. In addition we would recommend you read our Spark building methodology for help on thinking through problems at:

[http://www.sparkmapping.com/sparkmap/manual/spark\\_development\\_process.jsp](http://www.sparkmapping.com/sparkmap/manual/spark_development_process.jsp)

### **Managing authors in a production environment**

If you need information on creating author accounts, please refer to:

[http://www.sparkmapping.com/sparkmap/manual/organise\\_users.jsp](http://www.sparkmapping.com/sparkmap/manual/organise_users.jsp)

If you expect to have multiple authors working on the same server in a production or 'live' environment, we would recommend the following possible structures:

1. Each author has their own account
  - a. This allows the author to publish Sparks and work on updates before going live with the changes. Each author account is independent and will not affect what is happening in other accounts. The author is then free to distribute links to others for validation and checking / approval.
2. An author account is created for 'live' or production Sparks
  - a. The details for this account are only provided to key staff so any changes can be controlled and easily managed
  - b. This is the account that will be used to publish and manage all live hyperlinks – i.e. finished Sparks that you wish to make available to people to use
  - c. You can use this account to get live reporting and compliance information
  - d. The main company templates can be managed from this account

Using the above structure allows your organisation to manage change control and ensure development is kept away from the live Sparks. Equally, you can implement whatever change control you feel fit around this structure when going through a deployment cycle.

An example change control process might be:

#### ***Phase I - Draft***

1. The author creates the first draft Spark
2. The author distributes the Spark Map to others for review and comments
3. The author collates the comments and recommendations and makes the desired changes

#### ***Phase II – Test***

4. The author publishes the Spark to the server and sends the link page to all interested parties for review (we'll call them the testers)
5. Each tester goes through the Spark and provides notes as they go (using the Notes facility provided within Web Server Edition – this is a setting on the configuration page for the Spark: Sparks -> Update -> Provide notes facility)

6. The author reviews the interviews on the server and makes any changes as noted by the testers
7. The author publishes the changes to the server and iterates through this process as many times as required

***Phase III – Deploy***

8. The author gives the Spark to one of the key staff members that is able to publish into the live or production author account
9. The key member of staff sends the link page to all interested parties for final review and comment (usually done via e-mail / telephone / workshop)
10. The key member of staff creates the link page for the web development team to add to the necessary site page
11. Final check that the hyperlink and Spark are working as expected

## Setting up web design templates

Sparks can be branded in a variety of ways using web design templates. You can have many templates available in an author account to use with your Sparks. For information on creating templates, please refer to:

[http://www.sparkmapping.com/sparkmap/manual/organise\\_templates.jsp](http://www.sparkmapping.com/sparkmap/manual/organise_templates.jsp)

In addition to the reference material provided above, it is worth noting a few things:

- You can change the 'System Default' template by logging in as an Administrator. Simply update the 'System Default' template as required.
- You should keep the JavaScript functions that are included in the default template – if they are omitted, you will get JavaScript errors
- The style sheet at the bottom of the page gives you the near complete breakdown of the CSS parameters available to the web designer
- The content and navigation of your Spark is provided between the header and footer HTML so you have complete flexibility for the look and feel of the surrounding screen – use the available CSS parameters to control the look and feel of the content and navigation
- If you delete the text of a button, the button will not appear

## Techniques for linking to your Spark

When considering the process of linking to your Sparks, there are few key questions you need to answer:

- Do you want users to search for the Spark?  
As an example, you may have a knowledge base or other knowledge management technology you wish to integrate with Firefly
- Do you want to make the Spark navigation very simple?  
If so, it's often a good idea to have the Spark in a separate window so other website links are not confused with Spark navigation
- Do you want to offer users the option to stop-and-save?  
For example, they may need to go away and come back to the Spark at a later time

### *Adding a search capability*

Firefly does not have an in-built search facility, so we would recommend using your knowledge base, content management system or other knowledge management system to provide this functionality.

#### Example implementation

1. Create a short article in your searchable knowledge base outlining the details of your Spark
2. Create a hyperlink within this article that links directly to the Spark

#### Example user process

1. The user goes to your knowledge base and searches for a solution
2. The user gets a range of articles back, including one that outlines the details of your Spark
3. The user reads the short description and clicks on the link to start the Spark
4. The user goes through the Spark and gets advice on solving their problem

### *Simpler user navigation*

To make it easier for your users to navigate a Spark, we recommend opening the Spark in a new window with limited browser navigation controls. This allows them to focus on answering the questions without accidentally navigating into another section of your website accidentally.

#### Example implementation

1. Update the spark and tick the option for 'Close Spark window on finish' – this will close the window when the user finishes or saves
2. Create your link page from the author account on the web server
3. Put the link into a simple pop-up java script link, for example:  

```
<a href="javascript:
window.open('http://yoursparkurlhere','Spark','width=650,height=500,men
ubar=no,toolbar=no,location=no,directories=no,scrollbars=yes,resizable=yes')
;">Your link text here</a>
```
4. Add this link to your page as required (or your search system as specified above)

#### Example user process

1. The user finds the page with some information about your Spark
2. The user clicks on the link which opens a new, smaller browser window
3. The user goes through the Spark and gets advice on solving their problem

### ***Adding stop-and-save***

To allow stop-and-save, there are three possible options:

1. The user must log into the User Desktop (please see [http://www.sparkmapping.com/sparkmap/manual/the\\_user\\_desktop.jsp](http://www.sparkmapping.com/sparkmap/manual/the_user_desktop.jsp)) which contains functions and features for them to save and manage their Spark interviews
2. You must provide Firefly with a unique identifier to load the correct interview
3. Custom integration (please contact us for information on this)

To use the unique identifier approach, you must provide a unique ID (text and or numbers) that can be used to identify a particular interview. If you pass this into Firefly, the server will use the unique ID to identify the correct interview.

#### Example implementation

1. Create your link page from the author account on the web server
2. Add the following parameter to the end of the hyperlink:  
&uniqueID=**a unique identifier for the interview** (e.g. ...&uniqueID=12345)
3. Change the hyperlink location to:  
`http://yourserverurl.com:8080/sparkstudio2007/views/integration/index.jsp?...` replacing fireflyclient.jsp
4. Add this link to your page as required (or your search system as specified above)

#### Example user process

1. The user clicks on the hyperlink containing the unique identifier
2. The user goes through the Spark and stops-and-saves (perhaps closing the window)
3. The user performs other tasks
4. The user clicks on the hyperlink containing the identical unique identifier
5. The user continues through the Spark from where they left off, and gets advice on solving their problem

### **Additional Information**

If you would like information about how to improve technical call handling within your organisation please contact us at via our website [www.informavores.com](http://www.informavores.com) or call us in the US on +1 800 773 3061 or UK and other countries on +44 161 907 2567.