



Put Artist and Title on the Web

Ready-to-Use Solution

Introduction

The **Magic RDS 4** is a software for **FM radio stations** and **Internet radio stations** which incorporates a set of simple text publishing tools. Typically that text consists of artist and title of currently playing song which is provided by your broadcast automation system or just an mp3 player. That text can be redirected to your **website or any application and device** which accepts some of the connection types and protocols offered by the Magic RDS 4.

Multiple text sources and text outputs are supported, as well as text filtering and replacing. The text sources are switchable during the day, based on several conditions or a schedule.



Broadcast Automation



Magic RDS 4



Your Web Server

TIP

For evaluation and debug purposes, all functions and features are available in the Basic (free) license as well.

Implementation Guide

Following steps are required for getting it working:

1. Add a new connection

Adding a new connection to the Magic RDS 4 will create a bridge between the application and the destination server. In this example we will use simple HTTP GET query method for putting the text to the server. The Magic RDS 4 application may be either blank or other connections may already exist.

2. Create a new external text source

The external text source is a feature of the Magic RDS 4 whose purpose is reading of the text we want to publish. Several source types and processing options are supported, incl. local file read, http read, ftp read, plain text, XML, JSON, RegEx etc. Once the text changes or updates, it is forwarded to the Connection. A delay can be configured to sync the text with listener's audio.

3. Insert a simple php script to your web server (optional)

This step is necessary only if there's no plugin or function built-in which accepts, saves and publishes the text data.

Step 1 - Add a new connection (Bridge)

- Double-click on the "Add New Connection" icon
- Select Connection Kind: **Bridge**, Connection Type: **Internal Channel**, Connection Name: **Web Server**
- In the Data Distribution box, type following line:
`sendhttp url:"https://yourserver.com/show.php?s=%data%"`

where

yourserver.com is your domain and path to the script file which accepts the text

save.php is a script file which accepts the text and saves it either to a database or the web server local file. In some cases, the script file name is not required.

s is the variable name accepted by the server

%data% will be replaced by (URL-encoded) text in the final output. Type it as you see.

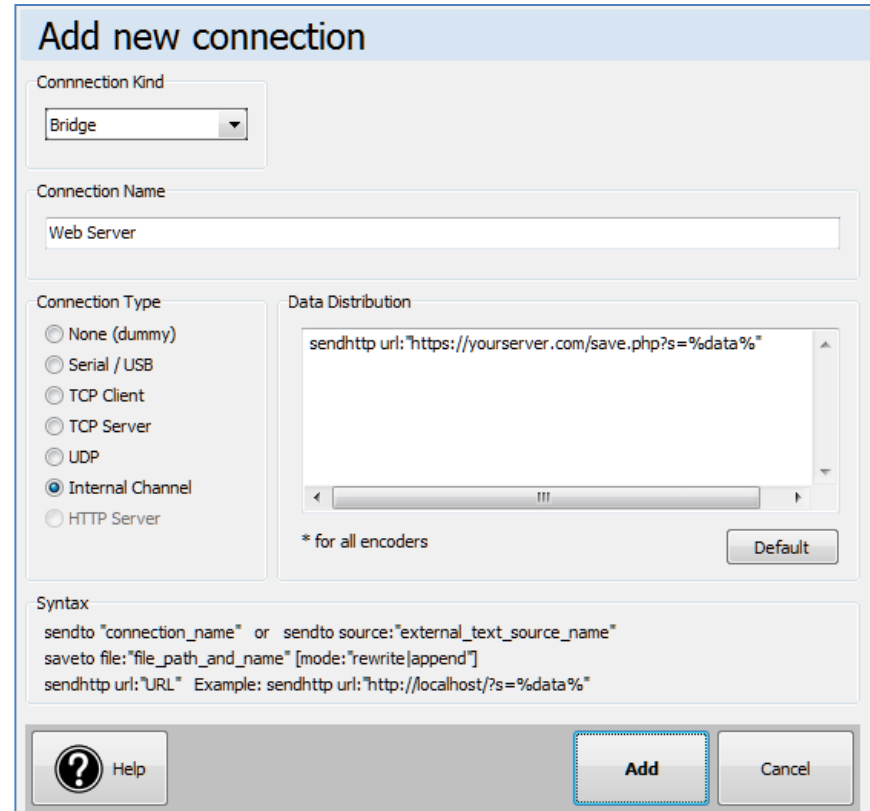
Contact your webmaster or follow your application/device manual for details.

TIP

You may fill more lines to the Data Distribution box if there's a need to send the text to more destinations (servers).

You may put another variable(s) to the URL string, with a fixed content, for example `&password=1234`

You may save the text to a local file as well (see the Syntax).



Add new connection

Connection Kind
Bridge

Connection Name
Web Server

Connection Type
 None (dummy)
 Serial / USB
 TCP Client
 TCP Server
 UDP
 Internal Channel
 HTTP Server

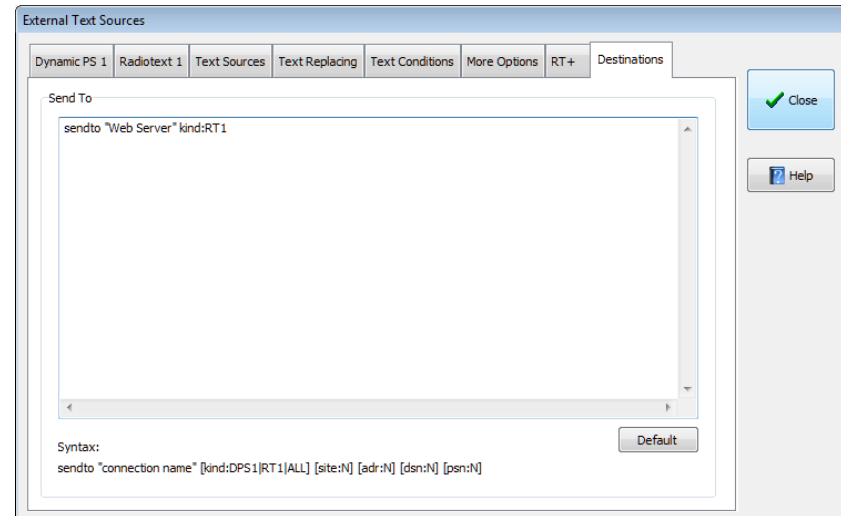
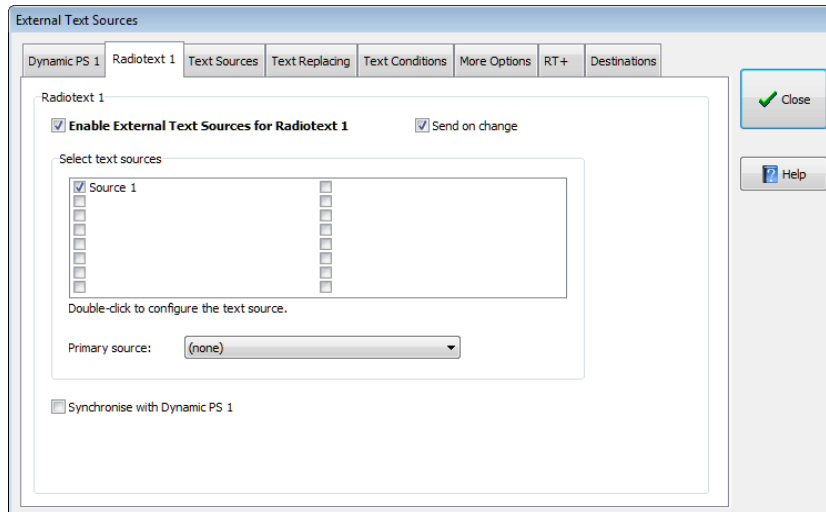
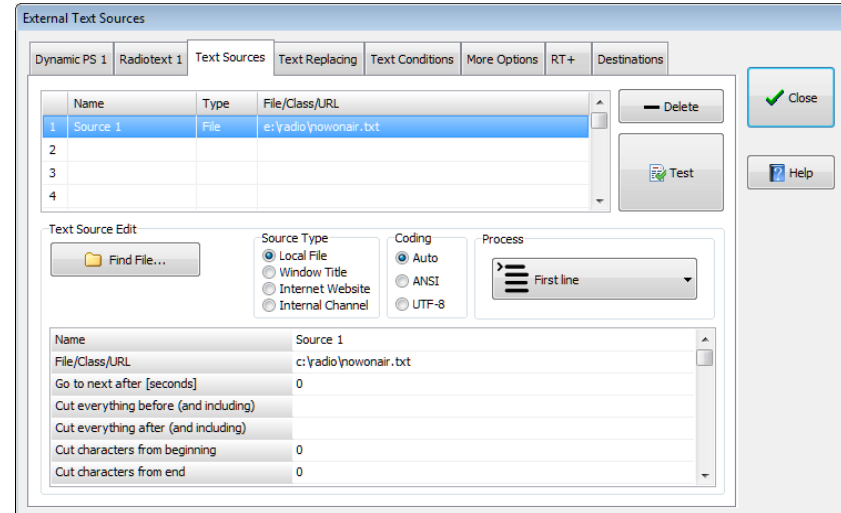
Data Distribution
sendhttp url:"https://yourserver.com/save.php?s=%data%"
* for all encoders
Default

Syntax
sendto "connection_name" or sendto source:"external_text_source_name"
saveto file:"file_path_and_name" [mode:"rewrite|append"]
sendhttp url:"URL" Example: sendhttp url:"http://localhost/?s=%data%"

Help Add Cancel

Step 2 - Create a new external text source

- Select Tools - External Text Sources
- Go to the Text Sources card and configure a new source (click on the Help button for details or see some examples on our website)
- Click on the Test button and check the result
- Go to the Radiotext 1 card and make sure the source is enabled and selected
- Go to the Destinations card and add new line to the box: *sendto "Web Server" kind:RT1* (according to the connection name put in previous step)



Step 3 - Insert a php script to your web server (optional)

This step is necessary only if there's no plugin or function built-in which accepts, saves and publishes the text data. Contact your webmaster for details.

TIP

For debug purposes, test the URL from the Step 1 in your web browser. Replace the %data% with any text manually.

The task can either be solved without or with a database:

Without the database, the text will be saved into a file. The \$filename must comply with the web server local file system and access rights.

To save the text (save.php)

```
<?php
$s = $_GET['s'];
$filename = 'show.txt';
if ($handle = fopen($filename, 'w')) {
    fwrite($handle, $s);
    echo 'OK: ' . $s;
}
fclose($handle);
?>
```

To show the text to your listeners

```
<?php
$filename = 'show.txt';
echo file_get_contents($filename);
?>
```

An example with database follows:

Before attempt to execute these scripts, make sure the database is accessible and appropriate structure is created, for instance by phpMyAdmin and SQL query:

```
CREATE TABLE listener_info(pkey varchar(255), pvalue varchar(255), PRIMARY KEY (pkey));
```

In the php files, above the sample code, enter these variables with correct values: \$host, \$usr, \$pwd, \$db

To save the text to DB (save.php)

```
$s = $_GET['s'];
$mysqli = new mysqli($host, $usr, $pwd, $db);
$sql = "INSERT INTO listener_info (pkey,
    pvalue) VALUES ('currentSong', '$s')
    ON DUPLICATE KEY UPDATE pvalue = '$s'";
if ($mysqli->query($sql)) {
    echo 'OK: ' . $s;
}
$mysqli->close();
```

To read the text from DB and show to your listeners

```
$mysqli = new mysqli($host, $usr, $pwd, $db);
$sql = "SELECT pvalue FROM listener_info
    WHERE pkey='currentSong'";
$result = $mysqli->query($sql);
if ($row = $result->fetch_assoc()) {
    echo $row["pvalue"];
}
$mysqli->close();
```