

## RCS Zetta® settings for DEVA Audio Processors RDS Console

**Zetta®** is software fully compatible with several devices from DEVA's Audio Processors product line:

- DB6400 - FM & Digital Radio 4-Band Processor;
- DB64-FM - Budget 4-Band FM Radio Processor;
- DB6000-STC - Stereo and RDS Generator.

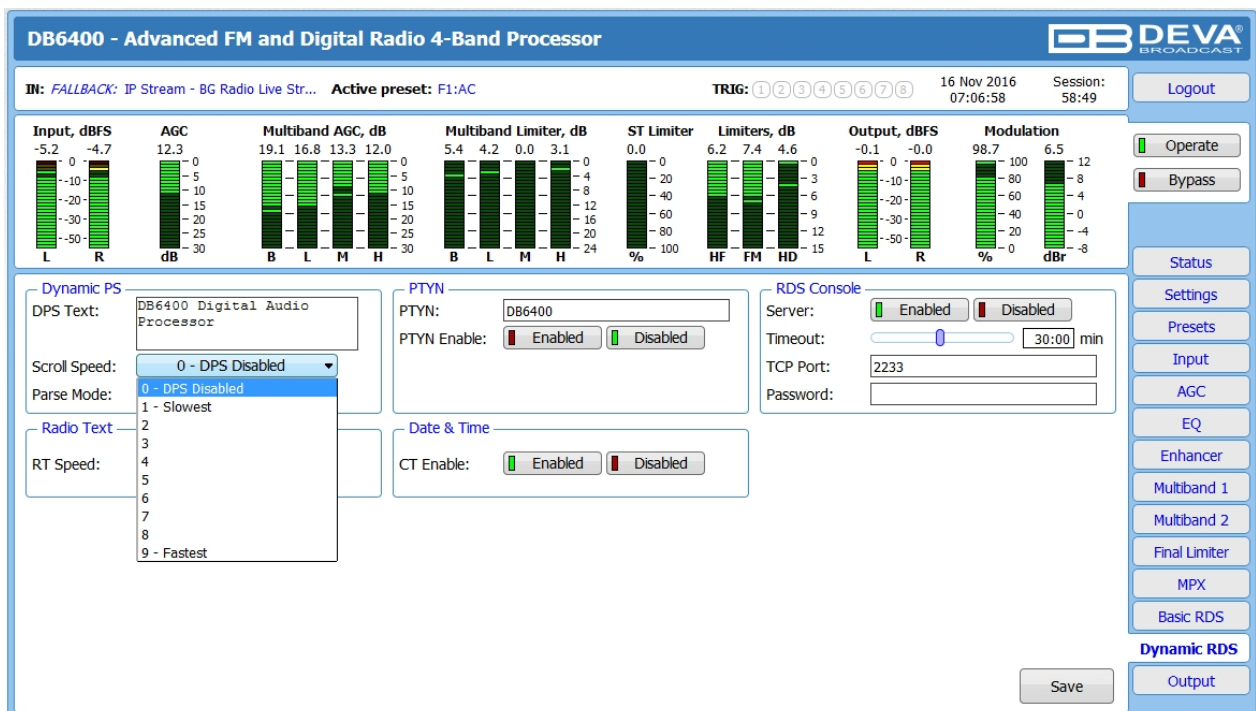
### 1. Set-up for the DEVA Audio Processor

1.1 Open the device's WEB Interface, then go to Dynamic RDS > section Dynamic PS.

1.2. DPS setup - Allow the usage of dynamic PS by setting Scroll Speed to a non zero value using WEB interface or the ASCII command DPSS.

Example:

**DPSS=5**

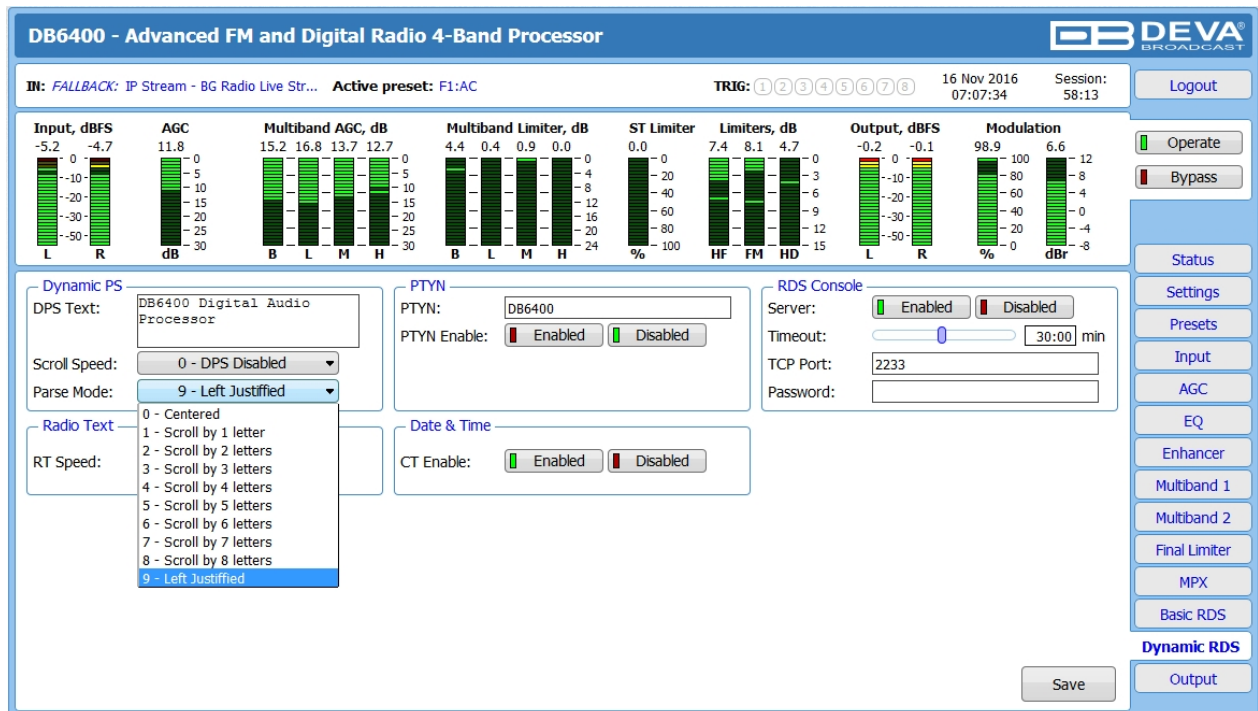


The screenshot displays the DEVA DB6400 web interface. At the top, it shows the device name 'DB6400 - Advanced FM and Digital Radio 4-Band Processor' and the DEVA logo. Below this, there are status indicators for the current stream ('FALLBACK: IP Stream - BG Radio Live Str...') and the active preset ('F1:AC'). The interface includes several real-time monitoring graphs for Input, AGC, Multiband AGC, Multiband Limiter, ST Limiter, Limiters, Output, and Modulation. The main configuration area is divided into sections: 'Dynamic PS' (with a dropdown for 'Radio Text' showing options 1-9), 'PTYN' (with 'PTYN Enable' set to 'Enabled'), 'Date & Time' (with 'CT Enable' set to 'Enabled'), and 'RDS Console' (with 'Server' set to 'Enabled', 'Timeout' at '30:00 min', 'TCP Port' at '2233', and a 'Password' field). A 'Save' button is located at the bottom right of the configuration area. On the right side of the interface, there is a vertical menu with buttons for 'Operate', 'Bypass', 'Status', 'Settings', 'Presets', 'Input', 'AGC', 'EQ', 'Enhancer', 'Multiband 1', 'Multiband 2', 'Final Limiter', 'MPX', 'Basic RDS', 'Dynamic RDS', and 'Output'.

1.3 Determine the appropriate mode of displaying of the dynamic PS text using WEB interface or the ASCII command PARSE. When PARSE is set to 0 (words centered) or 9 (words justified to the left), parsing will send the short words together. Long words (up to and including 8 characters) are sent individually/separately. Words exceeding 8 characters are "sidestepped" through two or more consecutive displays. When PARSE is set between 1 and 8, the message is scrolled from 1 to 8 characters at a time without dividing into word groups.

Example:

**PARSE=0**



1.4 In section RDS Console:

- *Server* – [Enable] the RDS console remote access;
- *Timeout* - specify session timeout. Upon expiration of the time set the connection will be closed;
- *TCP Port* – enter the TCP port of the RDS console. This console is used to edit RDS settings in real time. Default value is 2233;
- *Password* – Password identification support depends on the software. If your software does not support this feature, the field must be left blank. For further information on whether a password should be set, please refer to the complete user manual of the Automation Software, or the provider.

**WHEN APPLYING NEW SETTINGS** - In order new settings to take effect, it is necessary to press the [Save] button.

## 2. Zetta® software

To add new feed, you have to add a Now Playing Export from Zetta's System configuration. This is made via the Billboard tab that could be found in Zetta's system settings. The procedure is described in details below:

1. Clicking the **Add New Billboard** button will enable the following configuration fields:

- **Active** – Selecting the active option will enable the selected billboard instance;
- **Name** – Use the *Name* field to identify the billboard instance;
- **Description** – Use the *Description* field to provide additional information regarding the billboard service (optional);
- **Station** – Use the *Station* option to select the station that will send the Now Playing data from the drop-down menu;
- **Computer** - Use the *Computer* option to select the computer that will run the billboard service for the selected station.

The screenshot shows a configuration window for a billboard instance. It includes the following fields and options:

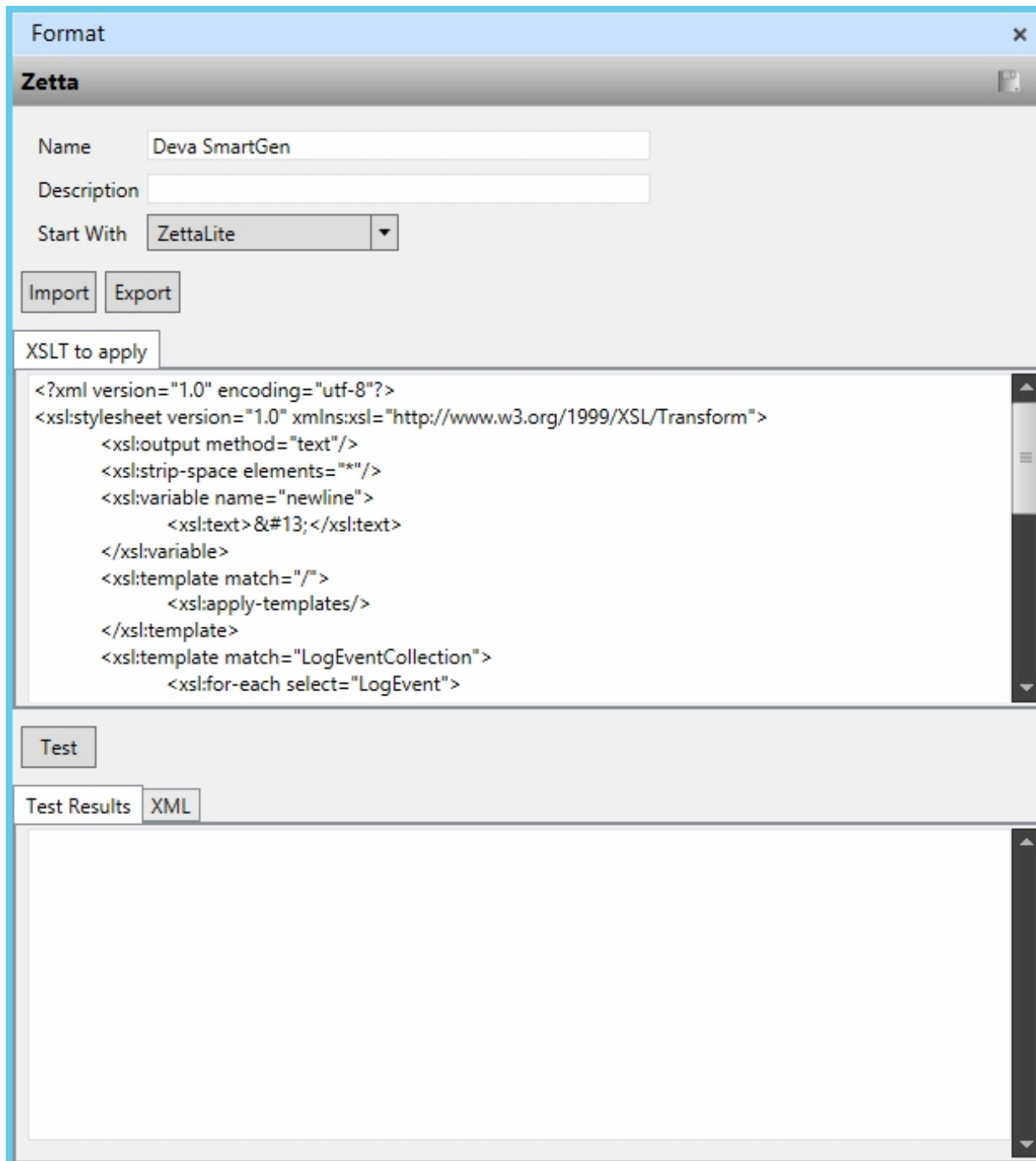
- Active
- Name: Deva SmartGen
- Description: Description
- Station: (Available)
- Computer: .zettaSatModeTest(Win8)
- Format: Deva SmartGen (with +, X, and edit icons)
- Number of Items: 1
- Data Encoding: UTF8
- Capture Now Playing Export output data
- Send changed message:  extra times
- Send single message every:  seconds
- Device: TCP
- IP Address/Computer Name: 127.0.0.1
- IP Port: 1024
- Connection Style: Close connection after each message

**NOTE:** Choose the correct Station, and a computer to execute this process.

2. Choose the communication type TCP/UDP or Serial. Depicted above is connection via TCP. Change the IP and Port to the appropriate one for the encoder.

3. The format should be added using the + button. Pressing the + button on Format provides the following screen, except that the XSLT to apply is blank. Press the Import button and then import the .xslt file you would like to be uploaded. Upon completion of the procedure, the window will look as depicted below.

The content of the xslt file used in this set-up is available in Step 4. Feel free to modify if you think there is a better way to format this than putting Title – Artist under „DPSTEXT=“. Carefully modify the text in the *Otherwise* section of the chooser. 2/3 down the screen to provide a good default message in the event of no data to display, if desired or blank it out. Press the Test button to insure that using test data your xslt is properly applied. Be sure to leave the “Start With” xml as ZettaLite.



The screenshot shows the 'Format' dialog box for 'Zetta'. It includes fields for Name (Deva SmartGen), Description, and Start With (ZettaLite). There are 'Import' and 'Export' buttons. Below is a text area for XSLT code, followed by a 'Test' button and a 'Test Results' section with an 'XML' tab.

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="text"/>
  <xsl:strip-space elements="**"/>
  <xsl:variable name="newline">
    <xsl:text>&#13;</xsl:text>
  </xsl:variable>
  <xsl:template match="/">
    <xsl:apply-templates/>
  </xsl:template>
  <xsl:template match="LogEventCollection">
    <xsl:for-each select="LogEvent">
```

For further information on **Zetta®**, please visit:  
<ftp://rcsworks.com/zetta/Zetta%20Help%20Guide.pdf>

#### 4..xslt Content

Below is the information that the xslt file should contain.

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="text"/>
<xsl:strip-space elements="*" />
<xsl:variable name="newline">
<xsl:text>#13;</xsl:text>
</xsl:variable>
<xsl:template match="/">
<xsl:apply-templates/>
</xsl:template>
<xsl:template match="LogEventCollection">
<xsl:for-each select="LogEvent">
<xsl:text>DPSTEXT=</xsl:text>
<xsl:choose>
<xsl:when test="@Type='SONG'">
<xsl:value-of select="Asset/@Title"/>
<xsl:text> - </xsl:text>
<xsl:value-of select="Asset/@Artist1"/>
</xsl:when>
<xsl:otherwise>
<xsl:text>Your Station Name Rocks!</xsl:text>
</xsl:otherwise>
</xsl:choose>
<xsl:value-of select="$newline"/>
</xsl:for-each>
<xsl:apply-templates/>
</xsl:template>
</xsl:stylesheet>
```