
DazzlerMax 5.5 Release Notes

Introduction

These notes are designed to explain the new features in DazzlerMax 5.5. The main focus of development has been to provide one-click support for the latest standards in e-Learning through the SCORM specification for interoperability with Learning Management Systems. There are also numerous other features to enhance the delivery of e-Learning with DazzlerMax

We recommend that if you are not already familiar with the SCORM standard itself you should visit www.adlnet.org.

Please note: DazzlerMax 5.5 will only be available in one version, "Deluxe", with the full range of variable handling needed for delivering e-Learning applications. There will no longer be a cut-down entry level version of DazzlerMax.

New features

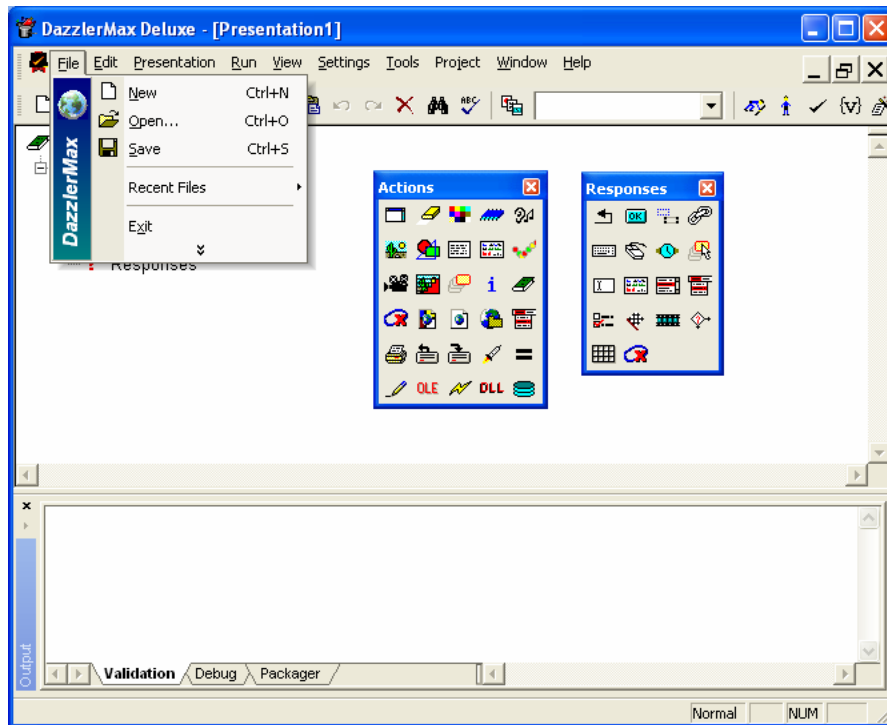
Here is a summary of the new features:

- There is also a new (optional) look and feel to the development system interface - it will be familiar to users of Microsoft Outlook 2000 etc. with 3D menus and effects and hidden, infrequently used menu items.
- Packaging for a SCORM LMS. The packager will automatically generate the XML wrapper and associated files needed for DazzlerMax courseware to be treated as a SCORM "SCO" and hence be launched and tracked by any SCORM compliant LMS. The content can also be loaded in a tool like Microsoft's LRN to examine the XML structures and to test the courseware and edit the manifest file to change the SCORM structure of the course.
- Communicating with an LMS. The Read and Write to File have now been extended so that they can also read and write data to a SCORM LMS, so student names can be passed in from the LMS and scores passed back, for example.
- Script Action - if running from CD-ROM etc. the Script will be assumed to be MCI, just like the old version. However, the Java player applet interprets the script as JavaScript and will make function calls to Java functions accessible on the webpage the applet is called from. Variables can be passed back and forth : e.g. `myVar = myJavaScriptFunction1({StudentName})`
- A user-defined Javascript file is also scanned to create a list of JavaScript functions for use by the Script Action.

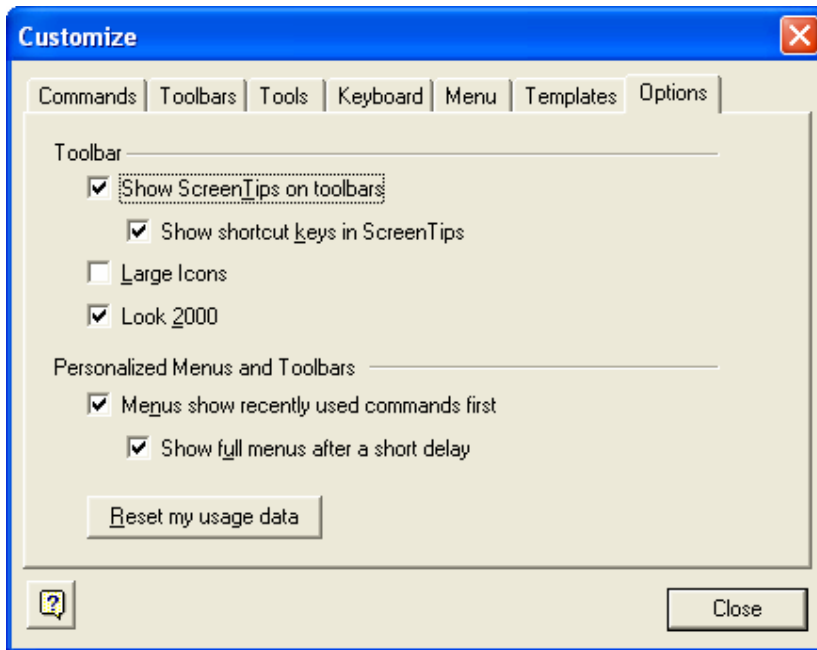
- The Java player applet now supports databases via the ODBC Action and JDBC drivers on the server. You set up an ODBC connection on the server with a JDBC driver like JDataConnect and then the Dazzler applet interprets the ODBC Actions you've put in your course.
- Additional information can be passed into the DazzlerMax Java applet to tailor its use within an e-learning environment.

New interface

We have redesigned the interface so the DazzlerMax 5.5 supports the new “Outlook 2000” style with a new menu look and unused menu items hidden. Menu animation effects are also supported. When running under Windows XP, the full new look is revealed, with new dialog box styles:

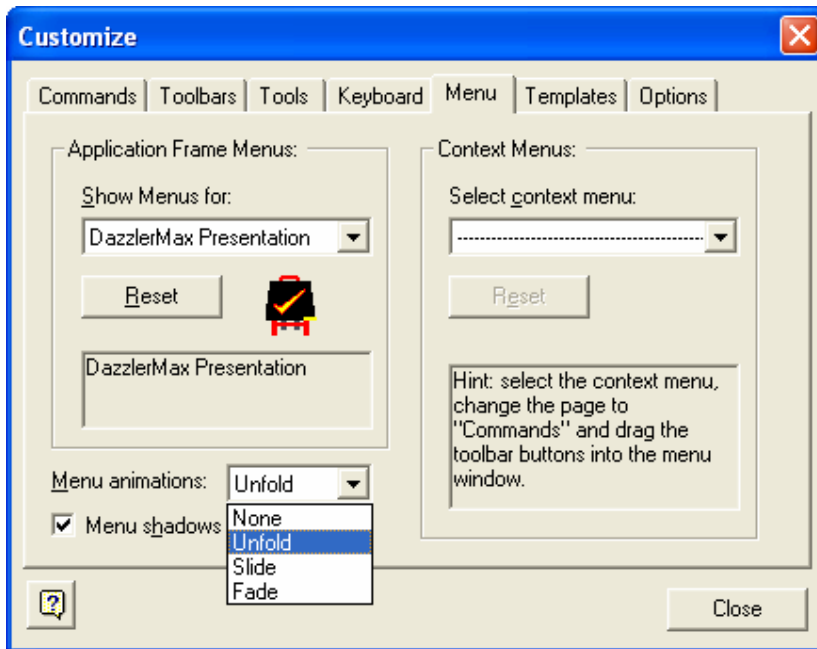


However, if you don't like this new style you can switch it off or on. Go to the *Settings* menu and select *Customize...* . Click on the *Options* tab:



Clear the *Look 2000* check-box to remove the 3D look to the menus. Clear the *Menus show recently used commands first* box if you want to see all the possible menu options at all times.

To change the animation effects used when a menu is selected in the DazzlerMax development environment, click on the *Menu* tab in the *Customize* dialog box:



Click on the *Menu animations* drop-down box and select the effect you want. You can also further tailor the look and feel of the development environment using options such as the *Menu shadows* check box. The *Context Menus* part of the *Menu* dialog tab lets you determine which items are going to appear when you right mouse click in certain parts of the development environment, for example the RTF Editor in the Rich Text Action. The names on the menu can be edited, menu shortcuts added or removed etc.

Packaging for a SCORM LMS

After much consultation, we have implemented the following method for making the courseware you create with DazzlerMax compatible with Learning Management Systems which support the SCORM 1.2 standard. Please visit www.adlnet.org for more details about this standard.

At the simplest level, for those customers who just want to deliver to a SCORM 1.2 compliant LMS without having to worry about the details of the specification, then there is a simple additional check box in the Packager Wizard. Follow the Package for Internet/Java Player route.



Click the *Make Course compatible with a SCORM LMS* box. That's all you need to do. However, underneath DazzlerMax does the following things:

It creates a file called `IMSMANIFEST.XML` which is the main file a SCORM 1.2 compliant LMS will look for with any courseware it is going to manage. This XML file lists all the files associated with the Dazzler course content (or "learning object") and lets the LMS know how it should launch the course. In the case of DazzlerMax this will be through the HTML file which references the DazzlerMax Java Player and top level DZL file. The XML file will also list all the other DZL files and media files, such as pictures, needed by the course.

By the way, this file can be loaded into Microsoft's LRN Editor. From there the course contents can be examined and additional XML data relevant to an LMS added, if required. This toolkit is available for free download from the Microsoft web site at: <http://www.microsoft.com/elearn/support.asp>.

The HTML file generated the Packager will also contain additional information to tell the DazzlerMax Java Player to switch on additional code to communicate with a SCORM 1.2 LMS. through a JavaScript "wrapper" file called `dazzlermaxscorm1_2.js` which gets included by the Packager. When the LMS

starts the DazzlerMax course by launching the HTML file, the DazzlerMax Java Player will call various functions in the JavaScript wrapper file to let the LMS know that it is starting. When the course finishes it will again let the LSM know that via the function calls in the JavaScript wrapper.

However, remember that you don't really need to know any of this to make your course SCORM 1.2 compatible. – just tick the *Make Course compatible with a SCORM LMS* box. All the files in the packager directory can then be given to the LMS or website administrator for inclusion in the LMS managed content library.

This JavaScript file also contains functions to let the DazzlerMax course get and send data to the LMS. See the notes on the Read/Write Action in the Communicating with an LMS:

Communicating with an LMS

The Read and Write to File Actions have now been extended so that they can also read and write data to a SCORM LMS, so student names can be passed in from the LMS and scores passed back, for example. The range of data which can be passed back and forth is defined in the SCORM specification itself. In effect this represents a language spoken by SCORM compliant LMSs and course content. Many of the “words” of this language are optional, so you will need to check which ones are supported by the LMS you are intending to use.

To make it easier to pick which data items you want to get from and send to the LMS, we've added code to DazzlerMax to display all the items possible in the drop down boxes in the Read and Write Actions. You can alter which ones are shown in the list by editing the CMI.INI file in the main DazzlerMax directory.

Here's an extract from the file:

```
cmi.core._children      =1
cmi.core.student_id    =1
cmi.core.student_name  =1
cmi.core.lesson_location =1
cmi.core.credit         =1
cmi.core.lesson_status =1
cmi.core.entry         =1
```

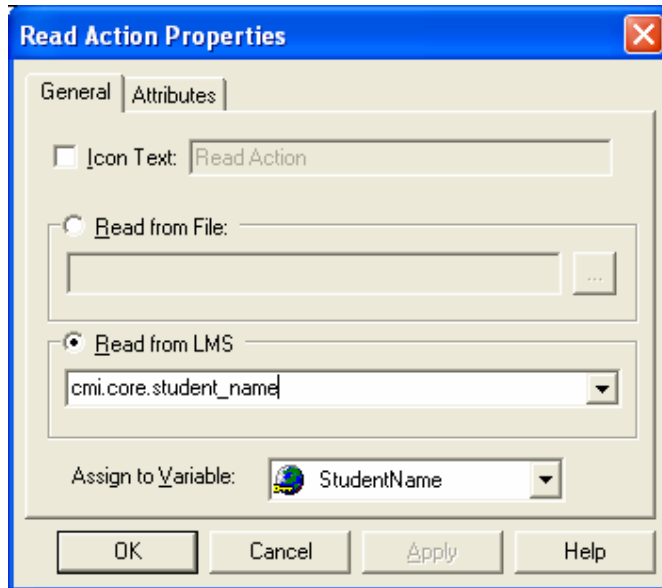
The “=1” part means that the associated SCORM data item e.g. cmi.core.children will be included in the drop down list of the Read and Write Actions property dialog in the DazzlerMax development environment. If you know that the LMS you will be using doesn't support particular SCORM data, you can stop it appearing in the Read/Write Actions drop down lists by editing CMI.INI in an editor like Notepad and just change the “=1” to “= 0” . For example, in the above extract, if you want to remove cmi.core.credit from the list, edit that line to read:

```
cmi.core.credit = 0
```

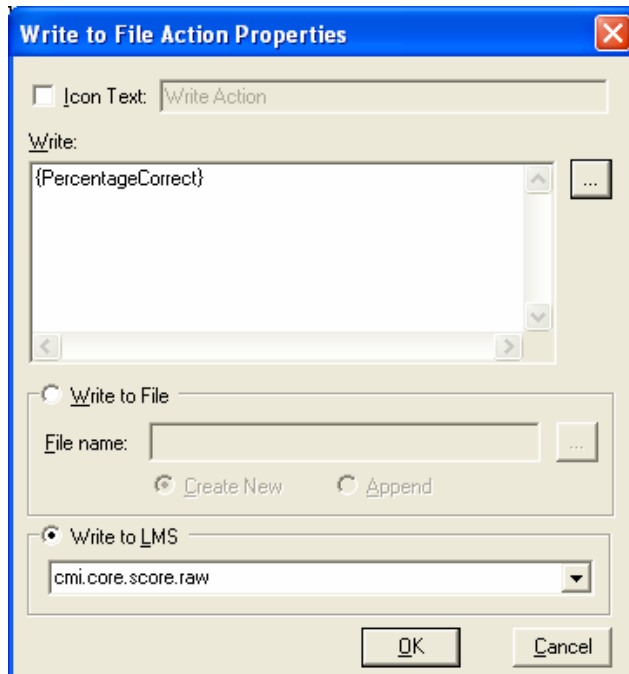
When you want to communicate with the LMS within your DazzlerMax course, for instance to request the name of the student taking the course so that you can display a tailored welcome screen within the course, use the Read and Write Actions. These Actions can now either exchange data with a file OR an LMS. To request the student's name from the LMS, drop a Read Action into your DazzlerMax course, double click on

it to bring up its properties dialog box and click on the bottom *Read from LMS* radio button.

Click on the drop down list beneath that and you should see a list of all the SCORM cmi... data elements. Choose cmi.core.student_name. You now need to tell DazzlerMax where it should store the information returned by the LMS. Click the Assign to Variable drop-down box and choose the StudentName variable, or any other variable where you want to store this information for use within the DazzlerMax course. When the course runs AFTER packaging from within the LMS, when it reaches this action the end result should be that the student's name, as stored in the LMS, is available in the DazzlerMax course, just as if it had been stored in a variable using the Variables Action such as StudentName = "John Doe".




The same mechanism applies to sending data to the LMS using the Write Action:




Script Action

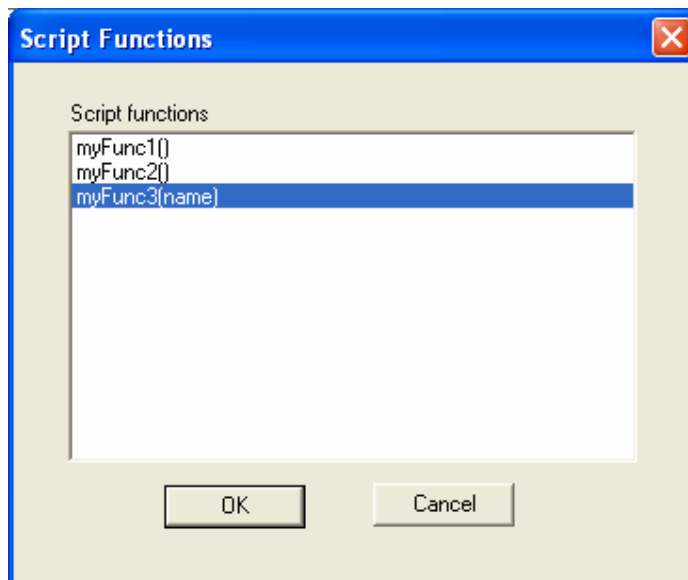
In DazzlerMax 5.5 you can now make direct calls to JavaScript functions so that, after packaging for the Java Player, your course can call JavaScript functions to extend the capabilities of the course. For example, at its simplest, you could call the *Alert* JavaScript function to display a message box at a particular point in a course.

Note that JavaScript functions only work after packaging for the Internet/Java Player. With any other option e.g. packaging and running from CD-ROM, for compatibility, DazzlerMax will assume the script is an MCI script for controlling multimedia devices, as it has always done.

The Script Action  (formerly the MCI Action) can call any JavaScript function which can be found by the DazzlerMax Java Player applet when it runs. This means any function which is either in the HTML page which calls the applet or is linked to it through a reference to a .js file in the HTML. To make the whole process easier, there is a file called `myfunctions.js` in the REDIST/JAVA directory of the installed DazzlerMax development system. You can edit this file to add in your own functions.

The file is scanned when you use the Script Action within the DazzlerMax development environment. Drop in a Script Action into your course and double-click on it to bring up its properties dialog box. To see a list available functions in the `myfunctions.js` file, click the browse button at the bottom right: 

A list of functions in the file will be shown



Highlight the functions you want to use and click OK. The functions will then be inserted into the Script box so that it can be edited for the values needed to be passed in. You can pass DazzlerMax variables into the JavaScript functions and return values from the functions back into variables.

e.g. `myVar = myJavaScriptFunction1({StudentName})`

Note that the lines in the Script Action are processed one by one, so you can use the values returned from one JavaScript function to pass into the next one called.

e.g.

```
myVar = myJavaScriptFunction1( {StudentName} )
```

```
myJavaScriptFunction2( {myVar} )
```

Note that variables passed into the functions need to be put in parentheses {...} whereas variables storing return values do not.

Databases and JDBC

In previous versions of Dazzler/DazzlerMax, it has not been possible to connect to databases from within a course packaged for the Internet/Java Player. Version 5.5 changes that through database support via JDBC technology. It works in conjunction with the ODBC mechanism familiar from the ODBC Action used in database functionality in DazzlerMax courses delivered via CD etc.

When the DazzlerMax Java Player applet comes across an ODBC Action in a DazzlerMax course, it will extract information from it, in particular the Data Source Name setup for the database in the *ODBC* part of *Control Panel* (or under *Control Panel/Administrative Tools/Data Source (ODBC)* in Windows XP. It will then try to open database connection to that data source on the web server using a JDBC (Java Database Connectivity) driver specified as an applet parameter in the HTML file which calls the DazzlerMax applet. DazzlerMax itself doesn't contain JDBC drivers, but there are many on the market for different types of databases which conform to this standard. There is a list of available drivers on Sun's website at <http://industry.java.sun.com/products/jdbc/drivers> . In our testing we have mainly used JDataConnect and the example here uses the settings for that driver

Two parameters needed to be added to the HTML which calls the DazzlerMax applet on the server:

```
<PARAM NAME = JDBCdriver VALUE = "JData1_2.sql.$Driver">
```

```
<PARAM NAME = JDBCurl VALUE = "jdbc:JDataConnect://" >
```

The values here will change depending on the JDBC driver used. The documentation for a particular driver will contain the relevant information.

You also need to make sure that you set up an ODBC data source on your web server with the same name as the one you've used on your development computer for testing via ODBC and ODBC actions.

Note: if you experience problems when using the database functionality under the Internet/Java player, please look in the Java Console of your browser since the DazzlerMax player will output lots of information about what it's trying to do and any errors which have occurred. It will also help us with technical support. In Internet Explorer the Java Console is not switched on by default. If it doesn't appear in the Internet Explorer *View* menu, switch it on as an option in the *Java* part of the *Tools/Internet Options/Advanced* menu.

Miscellaneous

Following requests from customers, we have extended the range of information which can be passed into the DazzlerMax Java Player applet via parameters in the applet call within the HTML page which launches it.

StudentName and StudentID can be passed in directly to the applet and hence stored and used in the internal DazzlerMax variables.

```
<PARAM NAME = StudentName VALUE = "Jane Doe">
```

```
<PARAM NAME = StudentID VALUE = "ABC1234">
```

This enables customers to pass data from other web applications, such as LMSs without having to go through the SCORM mechanism. Data can also be passed out from the applet using the HTML Action and specifying a target URL with which to communicate.

If your course is writing to a file from the Write To File Action via the DATAPUMP mechanism in the Java Player, then you can change where the data will be written by specifying it using another applet parameter:

```
<PARAM NAME = writeLocationURL VALUE =  
"http://www.mywebsite.com/dataoutput">
```

Similarly, if you want the Java Player to look for pictures and other resources in a location other than the one where the applet resides, use the following parameter:

```
<PARAM NAME = resourceURL VALUE = "mydata/pictures">
```

Note that the VALUE parameters for both of these can be a fully formed URL (e.g. <http://www.mywebsite.com/dataoutput>) or a relative path (e.g. [mydata/pictures](#)) where the applet will then expect to find the information it needs in a subdirectory of that name off the folder where that applet "dazzler.cab" etc. is stored on the server.

Contact

If you experience any problems or have any questions about this new release, please e-mail support@dazzlersoft.com or telephone +44 1273 670980.