

What's New in Omnis Studio 4.2

Raining Data Corporation

August 2006

The software this document describes is furnished under a license agreement. The software may be used or copied only in accordance with the terms of the agreement. Names of persons, corporations, or products used in the tutorials and examples of this manual are fictitious. No part of this publication may be reproduced, transmitted, stored in a retrieval system or translated into any language in any form by any means without the written permission of Raining Data.

© Raining Data Corporation, and its licensors 2006. All rights reserved.

Portions © Copyright Microsoft Corporation.

Regular expressions Copyright (c) 1986,1993,1995 University of Toronto.

© 1999-2006 The Apache Software Foundation. All rights reserved.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

OMNIS® and Omnis Studio® are registered trademarks of Raining Data Corporation.

Microsoft, MS, MS-DOS, Visual Basic, Windows, Windows 95, Win32, Win32s are registered trademarks, and Windows NT, Visual C++ are trademarks of Microsoft Corporation in the US and other countries.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world.

IBM, DB2, and INFORMIX are registered trademarks of International Business Machines Corporation.

ICU is Copyright © 1995-2003 International Business Machines Corporation and others.

UNIX is a registered trademark in the US and other countries exclusively licensed by X/Open Company Ltd.

Sun, Sun Microsystems, the Sun Logo, Solaris, Java, and Catalyst are trademarks or registered trademarks of Sun Microsystems Inc.

J2SE is Copyright (c) 2003 Sun Microsystems Inc under a licence agreement to be found at:
<http://java.sun.com/j2se/1.4.2/docs/relnotes/license.html>

MySQL is a registered trademark of MySQL AB in the United States, the European Union and other countries (www.mysql.com).

ORACLE is a registered trademark and SQL*NET is a trademark of Oracle Corporation.

SYBASE, Net-Library, Open Client, DB-Library and CT-Library are registered trademarks of Sybase Inc.

Acrobat is a trademark of Adobe Systems, Inc.

Apple, the Apple logo, AppleTalk, and Macintosh are registered trademarks and MacOS, Power Macintosh and PowerPC are trademarks of Apple Computer, Inc.

HP-UX is a trademark of Hewlett Packard.

OSF/Motif is a trademark of the Open Software Foundation.

CodeWarrior is a trademark of Metrowerks, Inc.

This software is based in part on ChartDirector, copyright Advanced Software Engineering (www.advsofteng.com).

This software is based in part on the work of the Independent JPEG Group.

This software is based in part of the work of the FreeType Team.

Other products mentioned are trademarks or registered trademarks of their corporations.

Table of Contents

ABOUT THIS MANUAL	5
WHAT'S NEW IN OMNIS STUDIO 4.2	6
WEB SERVICES.....	7
MAC INTEL.....	7
DIRECT HTTP ACCESS	7
WEB CLIENT	7
<i>Web Client for Windows Firefox</i>	<i>7</i>
<i>Drag and drop</i>	<i>8</i>
<i>Tooltips.....</i>	<i>8</i>
<i>\$startfield for remote forms</i>	<i>8</i>
<i>\$splitpathname for Formfile.....</i>	<i>8</i>
<i>Html templates.....</i>	<i>8</i>
LISTS	9
<i>List/Row subtypes</i>	<i>9</i>
<i>Defining Lists/Rows from file classes</i>	<i>9</i>
JAVA	10
<i>JVM properties</i>	<i>10</i>
<i>Java to Omnis data type mapping.....</i>	<i>10</i>
STYLES.....	10
<i>Copying styles.....</i>	<i>10</i>
<i>Printing styles.....</i>	<i>11</i>
OMNIS VCS.....	11
<i>Updating local libraries from the VCS.....</i>	<i>11</i>
<i>Show As Checked Out option.....</i>	<i>11</i>
DAMS.....	11
<i>MySQL DAM</i>	<i>11</i>
<i>Oracle8 DAM (Unicode only).....</i>	<i>12</i>
ENVIRONMENT VARIABLES	12
<i>Java Classpath.....</i>	<i>12</i>
<i>Font Location</i>	<i>12</i>
<i>Functions</i>	<i>12</i>
OMNIS DATA BRIDGE.....	13
<i>ODB Server preference.....</i>	<i>13</i>
FIND AND REPLACE	14
GRAPH2.....	14
DATA GRIDS.....	14
FILEOPS	14
<i>Unix File/Folder permissions</i>	<i>14</i>
\$SENDALL	14
<i>\$sendallref method</i>	<i>14</i>

Table of Contents

<i>\$sendall method</i>	15
SYS() FUNCTIONS.....	15
<i>sys(115)</i>	15
<i>sys(209)</i>	15
<i>sys(212)</i>	15

About This Manual

This document describes the new features and enhancements in Omnis Studio 4.2, and supplements the information provided in the Omnis Studio 4.1 manuals published in November 2005. Please see the file `Readme.txt` for details of bug fixes and any last minute notes in this release.

If you are new to Omnis Studio you should start by reading the *Introducing Omnis Studio* and then the *Omnis Programming* manual. All the Omnis Studio manuals are available on the product DVD and to download from the Omnis web site.

What's New in Omnis Studio 4.2

The following is a summary of the most significant features included in Omnis Studio version 4.2.

- ❑ **Web Services**
You can purchase a plug-in for Omnis Studio 4.2 that allows Omnis to act as a client or a server for Web Services; note that the Web Services plug-in is not supported in older versions of Omnis Studio.
- ❑ **Mac Intel**
Omnis Studio 4.2 will run on Apple computers containing the new Mac Intel processor as well as the older PowerPC chips. Note that Omnis Studio 4.2 requires version 10.4 of Mac OS X to run.
- ❑ **Direct HTTP access**
This release includes a basic HTTP server, incorporated into the Omnis executable, which is sufficient to handle Web Service and ultra-thin client requests sent directly to Omnis (note HTTPS is not supported).
- ❑ **Web Client**
In this version, the Omnis web client now allows drag and drop between fields in remote forms; plus \$tooltip and \$startfield are available for some remote form fields. In addition, the Web Client is now available for Firefox under Windows.
- ❑ **List/Row subtypes**
A schema, query, or table class name can now be used as the subtype of a list or row variable that is a class, instance, local, task or parameter variable, or a column in a list or row defined from a SQL class.

There are many other minor enhancements that have been added to the Omnis Studio core product that are documented here in this manual, including some that were added to support Web Services and Mac Intel compatibility.

Web Services

Alongside the release of Omnis Studio 4.2, there is a new plug-in component providing support for Web Services. You need the Web Edition of Omnis Studio 4.2, or higher, and you need to purchase a plug-in serial number (license) to use the new Omnis Web Services component. Please contact your local Sales office for further information about the Web Services plug-in or visit the Omnis website: www.omnis.net

Mac Intel

Omnis Studio 4.2 allows you to run your Omnis applications natively on any Apple computer containing the new Mac Intel processor or the older PowerPC processor. Mac OS version 10.4 (Tiger) is required in order to run Omnis Studio 4.2.

Direct HTTP access

Omnis Studio 4.2 includes a basic HTTP server, incorporated into the Omnis executable, which is sufficient to handle Web Service and ultra-thin client requests sent directly to the Omnis Server. The basic server does not offer HTTPS, and you may find that if your server has a high throughput, you still need to use a Web Server instead of the direct server.

To use the ultra-thin client with the direct server, use the following URL:

```
http://<server>:<serverport>/ultra
```

where <server> is the domain name or IP address of the machine running Omnis, and <serverport> is the value in the \$serverport property of the Omnis Server. For example:

```
http://127.0.0.1:5920/ultra
```

Web Client

Web Client for Windows Firefox

The Omnis Web Client is now available for Firefox on Windows, adding to the existing wide selection of browsers and platforms the plug-in already supports. The new web client plug-in for Windows Firefox requires version 1.5.0.5 of Firefox or above.

We have added the new plug-in to the Firefox Add-ons website which allows you to download and install the plug-in into your Firefox browser. Get the new add-on here:

<https://addons.mozilla.org/firefox/2997>

The plug-in will also be available to download on the Omnis website.

Drag and drop

Omnis Studio 4.2 allows drag and drop between fields in remote forms running in the Omnis web client. The drag and drop functionality is the same as for window class fields, except that drag object or duplicate is not supported. The drop mode for remote form fields can be either `kAcceptAll` or `kAcceptNone`, that is, none of the other `kAccept..` constants are valid for remote form fields. Drag and drop events must be processed by a `$event` method marked as 'Execute on client'.

Tooltips

You can now add a `$tooltip` to some Remote form objects, such as single-line edit fields and heading lists. When the user places their mouse over such objects the tooltip is displayed in the web browser.

`$startfield` for remote forms

The `$startfield` property, which specifies the first field in a form, is now available for remote form fields. For positive values, it is the order number of the initial enterable field of the remote form. If you specify a negative value, and a subform is created from the remote form class, the field specified in `$startfield` becomes the current enterable field on the form.

`$splitpathname` for Formfile

The `$splitpathname()` method has been added to the `FormFile` web component so now you can split a pathname entered by the client. The new method behaves in the same way as the `FileOps.$splitpathname` method.

Html templates

The Html templates to enable the Omnis web client plug-in have changed. Therefore, some of the information about the templates in the *Omnis Programming* manual is out-of-date. The changes are as follows.

In recent versions of Internet Explorer for Windows XP (ones with the latest security update from Microsoft, issued after April 11, 2006) when you open IE to use the web client, the Active X appears with a fuzzy border, and a tooltip telling you how to activate the control. This is due to a patent infringement lawsuit between Microsoft and Eolas Technologies which Microsoft lost.

There is a work-around however, which is to put Javascript generating the Active X control in a separate file. We have done this, together with rationalising the auto-download script in our various HTML pages. We have removed support for Netscape 6, and integrated support for downloading the Active X CAB into `template.htm`; now you need to set a variable in `template.htm` to turn on the CAB download.

Therefore a new `'template.htm'` has been placed in the `'html'` folder of the Studio development tree, and is accompanied by `'omniswebclient.js'` which must be present when

you deploy your web application. Note the additional 'download.htm' and 'java.htm' files no longer appear in the 'html/templates' directory in the Studio tree.

Existing users will note that the parameters in the HTML template that embed the Omnis web client (the ActiveX or Netscape-type object) have changed. For example, the parameter OmnisServer is now oserver.

Web Client and Server compatibility

Note to existing Omnis users To take advantage of any new features in the Omnis web client, you need to upgrade your Omnis Server and the end user's plug-in. In general, the majority of features introduced into the Omnis web client require new code in the web client plug-in itself (installed on the client machine) as well as in the Omnis executable installed on the Omnis Server. Therefore, both need to be updated when you use new functionality in any Omnis web-based applications.

Lists

List/Row subtypes

A schema, query, or table class name can now be used as the subtype of a list or row variable, that is, a class, instance, local, task or parameter variable, or a column in a list or row defined from a SQL class.

In addition, the schema class has a new property \$createinstancewhensubtype that controls whether or not there is a table instance associated with a List or Row variable with a schema class as its subtype; you can set this property in the schema editor – it defaults to kTrue for existing and newly created schema classes. When using the schema class exclusively with Web Services, it is likely that the table instance will not be required, and in this case turning off \$createinstancewhensubtype will therefore improve performance.

Omnis uses the subtype class to define the list or row, or in the case of parameters, to identify the expected definition of the list or row, although Omnis does not do anything if the definition does not match.

Defining Lists/Rows from file classes

The \$define() and \$redefine() methods define the columns for a list or row variable. Rather than using a variable to define a column, you can use a file class name (a string) from which all columns are used. Now you append /S to the file class name to skip empty columns in the file class.

Java

JVM properties

There is a new core Java preference, \$javaother. You can use this property to specify any other preferences or switches you wish to pass to the Java Virtual Machine, e.g. to set a system property value, set \$javaother to `-Dproperty=value`. Like the other Java preferences, it only affects the JVM if \$usejavaoptions is kTrue.

Java to Omnis data type mapping

There are also some Java performance improvements (including a new preference). There is a new Boolean library preference called \$javareturnsnative. The property is part of a performance improvement to the way in which arrays of Java data types are returned from Java to Omnis. When the property is set to kTrue, Java object methods called from the library return native Omnis types if a suitable conversion exists, rather than an object reference to a Java object. For example, a method returning a Java String object returns an Omnis Character value. Setting the value to kTrue also benefits from the improved performance.

Existing libraries have the \$javareturnsnative property set to kFalse for backwards compatibility, but all new libraries will have a default value of kTrue.

Note: If you wish to set the \$javareturnsnative property to kTrue in an existing library, you must change the types of the Omnis variables used to return values from Java. For example, when a Java method that returns a `java.lang.String` is called from an Omnis library with \$javareturnsnative set as kFalse, the Omnis return type will be Object reference. Changing the \$javareturnsnative property to kTrue means that the return types for these methods need to be changed to the Omnis Character type.

The Java to Omnis data type mapping is documented in the *Java Objects* chapter of the *Omnis Programming* manual.

Styles

Copying styles

A new method called \$copystyle has been added to enable you to copy styles between libraries (or the same library). \$copystyle(`rItem`[,`cNewName`]) copies the style to `rItem`, which is an item reference to the library, and renames if `cNewName` is supplied. The method returns kTrue if successful. For example:

```
Do $clib.$fieldstyles.TestStyle.$copystyle(  
    $libs.TargetLib, "myNewStyle")
```

Copies the TestStyle from the current library to TargetLib and renames it to myNewStyle. If the new name is not supplied, the original name is maintained. If the style already existed under that name, it will be overwritten.

Printing styles

You can now print a list of styles in the #STYLES system class by right-clicking on the class and selecting the Print Class option. The report shows a complete list of properties for each style in the #STYLES table.

Omnis VCS

Updating local libraries from the VCS

There is a new feature in the Omnis VCS that allows you to bring a local library up to date with the latest version of the library in the current VCS repository. If you have large libraries and multiple developers you may find this a quick way of ensuring that a local copy of a library has all the changes made by other developers.

To use this new feature, select a library in the Studio Browser and click the 'Update from VCS' option. Providing that you have a VCS session open and that there is a project in the VCS repository that matches the name of the current local library, this option will bring up the checkout window (in Copy Out mode only) listing all the classes that have a newer class modified date in the VCS than your copy of the library. It will also list all classes that have been added to the repository that do not exist in your local library. If there are classes which you do not want to update, you can uncheck them in the checkout window.

Note: All existing repositories should be updated first using the UpdateVCS library that is located in the /convert folder.

Show As Checked Out option

The Studio Browser has a new context menu item called 'Toggle Show As Checked Out'. This toggles the \$showascheckedout property for the selected classes, so is faster than using the Property Manager to change the property for a each individual class.

DAMs

MySQL DAM

There is a new Session method for the MySQL DAM (DAMMYSQL).

\$queryresult()

SessObj.\$queryresult(IResult) allows the result set generated by a previous call to \$query() to be returned. IResult is a list variable which is cleared and redefined from the columns of the result set. The entire result set is placed into IResult and returned via a single call to \$queryresult(). \$queryresult() performs limited data type conversion on the various datatypes, recognising binary, integer and decimal numbers, defaulting to Character for all other types. This method has no effect if there is no result set pending on the session object.

Oracle8 DAM (Unicode only)

There is a new Session property for the Oracle8 DAM (DAMORA8), available in the Unicode DAM only.

❑ \$nationaltonclob

\$nationaltonclob is used to alter the default mapping of Omnis Character and National types. By default, Omnis Character and National fields with a subtype greater than \$maxvarchar2 are mapped to the NCLOB datatype. By setting \$nationaltonclob to kTrue only National fields with a subtype >\$maxvarchar2 are mapped as NCLOBs. Character fields with subtype >maxvarchar2 are mapped as non-Unicode CLOBs. Character fields mapped in this way are subject to data loss/truncation where such fields contain Unicode characters.

Environment Variables

Java Classpath

It is now possible to use the OMNISCLASSPATH environment variable to specify Java classes instead of CLASSPATH. This change should increase performance in situations where you have multiple Java applications installed that are not being used with Omnis. Under normal circumstances, the classes for these applications will be added to the Omnis Class Cache, even though they are not being used. By using OMNISCLASSPATH, you will be able to specify only the classes that you wish to use with Omnis.

If OMNISCLASSPATH is not defined, CLASSPATH will be used, thus ensuring backwards compatibility.

Font Location

There is a new environment variable called OMNISXIDIR which can contain the path where the OMNISXI.INI and OMNISXI.FCI are stored. For example, you could set the variable using:

```
SET OMNISXIDIR=/tmp
```

Functions

There are three new functions for manipulating environment variables.

listenv()

Returns a 2 column list of the Omnis process environment variables. Column 1 contains the variable name, and column 2 the variable value. The list is sorted by the variable name column (case insensitive). There is no need to define the list first; the returned list has 2 character columns, name and value. For example:

```
Calculate lList as listenv()
```

```
; returns a list of environment variables on the current system
```

putenv()

`putenv(name,value)`

Sets the Omnis process environment variable with the specified *name* to the specified *value*. Creates a new environment variable if necessary, and returns true for success, false for failure. For example:

```
Do putenv("OMNISCLASSPATH","C:\Program Files\RainingData\OS42\java")
; Creates the environment variable OMNISCLASSPATH and sets its
  value to C:\Program Files\RainingData\OS42\java
```

getenv()

`getenv(name)`

Returns the value of the Omnis process environment variable with the specified *name*. For example:

```
Calculate lVar1 as getenv("OMNISJVM")
; Returns the location of the JVM, e.g.
  C:\j2sdk1.4.2_11\jre\bin\client\jvm.dll
```

Omnis Data Bridge

ODB Server preference

As an alternative to specifying the IP address and port name in the *Open/Prompt for data file* commands, the *Set hostname* command, or the Hostname field in the SQL Browser, you can now store the IP address and port name of the ODB server in the Omnis preference \$odbserver (\$root.\$prefs) and omit the IP-address:port from the command parameters. The preference can contain the TCP/IP address and port of the ODB server in the form IP-address:port, e.g. 194.131.70.222:5900. To use this property, you write code such as:

```
Prompt for data file {odb://}
Open data file {odb://mydatafile}
  or
Set hostname {odb://odb_test}
```

The commands use the server identified by \$odbserver. In other words, the address and port components of the odb://datafilename string are not required if you use the \$odbserver property.

The value of \$odbserver is stored in a text file in the Studio folder, called odb.txt. This has one line, in the form:

```
server=address:port
```

This allows you to configure clients by either copying this file, setting up this file in an installer, or using the notation to set \$odbserver via your own dialog.

If the \$odbserver preference is specified, the Data File Browser includes an “Open ODB Datafile” option which will open the data file on the specified ODB server.

Find and Replace

The Find and Replace log context menu has two new items: Sort by Log Order, and Sort by Type. The log does not remember how it was last sorted when new items are added to it.

Graph2

The Graph2 component now supports printing. This includes the \$deviceindependent property supported by the old Graph component.

Data Grids

The data grid has a new property, \$showbinarylength. If true, the data grid displays the length of binary data, and also allows the droplist button for a cell to open a modal window to edit list, row and binary columns.

FileOps

Unix File/Folder permissions

The function \$setunixpermissions() has been added to the FileOps package. The function takes the file name as parameter 1 and the permissions in parameter 2, in the format returned by \$getunixpermissions. For example:

```
Do FileOps.$setunixpermissions('/omnis/myfile', '-rwxr-xr-x')  
; meaning that the file is only writable by its owner, and readable  
and executable by all users
```

Note that the \$setunixpermissions() and \$getunixpermissions() functions are available for Unix as well as Mac OS X.

\$sendall

\$sendallref method

A new method called \$sendallref() has been added. When using \$sendall, you can normally use \$ref to refer to the group member receiving the message. Now you can use \$sendallref, which is an item reference to the item currently receiving the message sent with \$sendall (note that \$sendallref is not supported in client methods). Consider the case where a parameter passed to the message is evaluated by calling another method, or a function implemented in an external component. In this case, if you use \$ref in the parameters passed to this other method or function, it will actually refer to the item involved in making the call to evaluate the parameter. This is where \$sendallref could be used, if you wish to

pass some property of the group member receiving the message to the other method or function.

For example:

```
Do $cinst.$bobjs.$sendall(
    $ref.$text.$assign(
        StringTable.$gettext(
            $class.$bobjs.[$sendallref.$ident].$text)))
```

The example uses the text stored in the class as the row id in the string table, and assigns the text stored in the string table to the background object. In the example, `$sendallref.$ident` returns the ident of the background object receiving the message. If you were to use `$ref.$ident`, the `$ref` would refer to the custom attribute representing the external component function, and the call to `$sendall` will not have the desired affect.

\$sendall method

The `$sendall()` method has a new optional third argument `bIgnoreUnrecognizedCustomAttribute`. This argument, which defaults to `kFalse` if omitted, causes `$sendall` to ignore unrecognized custom attribute errors, which would otherwise cause a runtime error when the library preference `$reportnotationerrors` is `kTrue`.

sys() functions

sys(115)

Under Mac OS X, `sys(115)` now returns the path to the folder containing the Omnis executable within the Omnis.app bundle, that is, `Omnis.app:Contents:MacOS:`

sys(209)

There is a new `sys(209)` function which returns the path to the Omnis.app bundle, regardless of the information returned by `sys(115)`. Omnis developers who currently use `sys(115)` and do not wish to place their libraries inside the Omnis.app bundle should switch to using `sys(209)`. On other platforms, `sys(209)` returns the same result as `sys(115)`.

sys(212)

There is a new `sys(212)` function which returns the list of sort fields for the executing method stack. The list has the following columns:

name	the name of the sort field
descending	boolean, true if this field is sorted in descending order
upper	boolean, true if sorting of this field is case-insensitive
subtotals	boolean, true if subtotals are printed when the sort field changes
newpage	boolean, true if a new page is started when the sort field changes