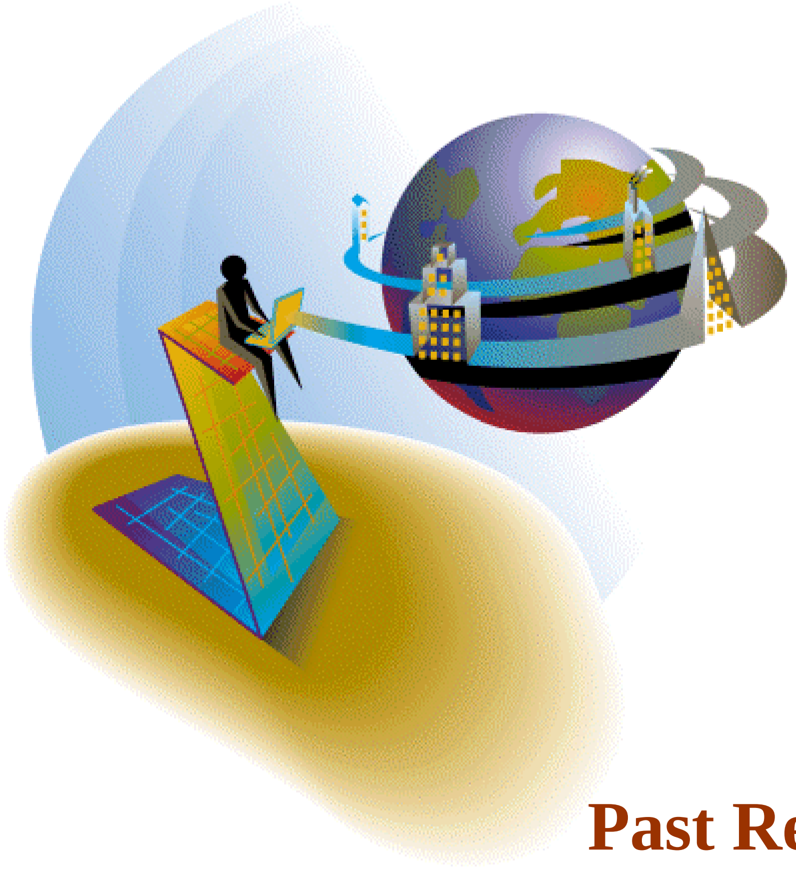


eDeveloper Past Release Notes

This document includes the major points provided in the release notes of all the minor versions and service packs of eDeveloper that preceded the last release of eDeveloper 9.4 Service Pack 6b.

For information about eDeveloper 9.4 Service Pack 6b release, refer to the ReleaseNotes.pdf file provided with this installation.

eDeveloper 9.01

Migrating to eDeveloper 9 – Changes of Behavior

General

Any Magic version 8 application can be exported in version 8 and imported to Magic version 9.

Please refer to the following changes in the Magic tool, engine and environment before you proceed with your migration to Magic version 9.

Java Forms

The Java form interface type has been removed from the supported interface types. Note that upon importing your Version 8 application, any Java form entry will be removed.

Magic Icons

Magic uses and provides a new set of Icons. Applications that use the Magic version 8 internal Icons will show different icons in version 9.

You should either use external icons or re-assign your menu entries to a new icon provided by Magic.

Color and Font files

The Magic eDeveloper provides a new color and font files. These files have been changed in structure so that the first 100 entries are for the internal use of Magic or for default settings of automatically generated objects. This way there is a clear differentiation between Magic internal colors and fonts and the user defines colors and fonts.

This change requires a conversion of the color and font references in the application.

In the following location you may find a utility application that will convert your colors and fonts references to work with the new color and font files.

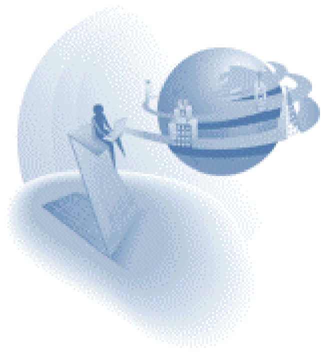
http://ftp.magicsoftware.com/v9product/Utilities/colorfont.zip

Event Handling

Magic version 8 task events are converted to call operation under a corresponding handler of Magic eDeveloper. (Application events are converted to Main program handlers).

Note that unlike Magic version 8, programs called from a high level handler cannot view or manipulate the variables of the task that initiated the event.

For more information please refer to the Magic Reference book, The Magic Application Engine chapter.



LEVEL function

The LEVEL function of the Magic eDeveloper returns different values than it used to in version 8. The effecting change of this function is that when referred to a Group level (On Change level in version 8) it returns GP_varname for Group Prefix and GP_varname for Group Suffix (as opposed to L1S and L2P in version 8).

Please refer the Magic reference book to learn more about this function.

Direct SQL statement of MSSQL

Constant strings that are used in DSQL statement should now be entered between two single quotes and not double quotes.

Reconnection capabilities

The reconnect capability of the gateways as provided in version 8 are not yet available in this release.

Transaction locking

The Magic eDeveloper introduces the concept of Physical and Deferred transaction modes. To make things clear, here is the default locking behavior for the Magic SQL gateways in each of the transaction modes:

Physical transaction mode:

All SQL gateways (except ODBC and Cache) use physical locking as their default locking strategy.

ODBC and Cache always use logical locking, regardless of the transaction mode.

Deferred transaction mode:

All SQL gateways use logical locking as their locking strategy.

Note: In order to maintain the previous locking behavior of Magic version 8 in MSSQL, the flag SQL_PHYSICAL_LOCKING=N should be specified in the Database Information field of the database property."

Report Generator

The Report Generator has been removed for the time being from the Magic engine.

Archive Manager

The Archive Manager has been removed for the time being from the Magic toolkit.

Application Automatic Modifications

The Magic eDeveloper redefined and improved some of its previously existing capabilities.

This makes various application definitions to be defined differently.

In most cases (all cases except the changes defined above) Magic will convert your application upon importing it from a previous version.

Any change made to application definition will be noted in the application import log file (XXIMP.LOG)

The intention of these changes is to maintain a similar flow of the application while adhering to the new structure of the application file.

Importing Magic eDeveloper V9.00 applications

General

Version 9.01 of Magic eDeveloper provides a new and better implementation of the Magic component framework. In the initial 9.00 version of Magic eDeveloper each object that was published as part of a component was identified by its internal ISN. (ISN is the internal identifier of each object in the application). The ISN of a Magic object is given implicitly by the Magic toolkit and it is not visible and manageable by the Magic developer.

Magic eDeveloper of version 9.01 will allow you to define your own public identifier by which a Magic object can be published as part of a Magic component. A public name column for each entry in each repository (including the Main program's events table) has been added. Providing a public name to a Magic object makes such an object available to be published as part of a Magic component.

Note that if you have already utilized the Magic component framework in the initial version of 9.00 a simple manual modification of the container application export file and providing public names to the published components objects are required as follows:

1. Export the applications that are used as components and import them as they are in version 9.01.
2. In each repository of these applications provide a public name to every object you have published.
3. Export the container application and before importing it to version 9.01 you should change the ISN of each object with the public name of the relevant object as described below

At the top of the export file you will find the entries of the various components loaded in your container application.

In this exported component repository section, the public names assigned to the Magic Objects in the Component MCF, should be added to COMP_ITEM description (it should be performed according to the existing order of the items in the component repository).

Example:

```
COMP={ISN=1,NMLEN=22,DSCLEN=55,COMP_TYPE=77,HDTXT="General ToolT
COMP_ITEM={ITEM_ISN=1,ISN=1,ITEM_OFS=101,PUBLIC="LoginInfoKey"},
COMP_ITEM={ITEM_ISN=2,ISN=2,ITEM_OFS=101,PUBLIC="AddressBUT"},
COMP_ITEM={ITEM_ISN=3,ISN=3,ITEM_OFS=101,PUBLIC="SeatsBUT"},
COMP_ITEM={ITEM_ISN=4,ISN=4,ITEM_OFS=101,PUBLIC="DeleteBUT"},
COMP_ITEM={ITEM_ISN=5,ISN=5,ITEM_OFS=101,PUBLIC="CreateBUT"},
```

The modified export file should be now imported to Magic eDeveloper of version 9.01

After these modifications the container application and the components applications should work correctly.

Persistent Browser Client Module

The Java applet that is the main part of the browser client engine module must be loaded by the browser client. The download time of this module can be considerable for some browsers. To address this issue, the Magic application server lets you make the Java applet module persistent on the client side. Once the applet is loaded for the first time, the applet is stored on the client machine. Then, whenever an additional browser task requires this module, the applet is taken directly from the client and does not need to be downloaded. This option requires end-user confirmation, using the browser built-in confirmation dialog, before the module is stored locally.

Browser Task Launcher

The initial response to a request for the browser task is a small HTML page that comes with a small launcher applet. This applet loads the browser client module to the client and re-issues the link back to the application server. If the browser client module is already persistent on the client side, the launcher immediately re-issues the request back to the application server, informing it about the location of the local browser client module. The launcher module file, MGLauncher.cab, should be placed in the same directory as the browser client modules.

The response to the re-issued request results in the final HTML page, which includes the interface definition and the XML information of the data and logic.

Because the launcher module places the browser client module locally, it requires end-user confirmation. If the end-user does not confirm this, the launcher does not preload the browser client module and directs the application server to have the applet loaded with the resulting HTML page. This is done using the URL of the embedded browser client module, which points to the applet that resides on the web server.

If the user accepts the launcher certification, an additional confirmation will be required for the actual browser client module.

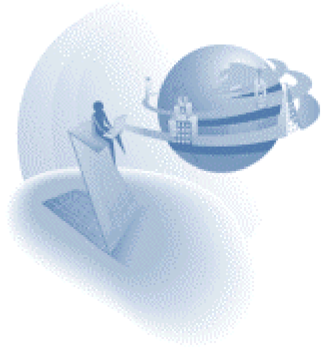
If the user chooses to trust the provider of the browser client module, Magic Software Enterprises Ltd., no additional end-user confirmation will be needed.

Signed Browser Client Module

The Browser client Java applet module has been certified and digitally signed by VeriSign®. This means that the applet code is verified by the VeriSign® authentication certificate, which provides the end user with the confidence that the content of this module has not been tampered with.

The signed applet file is different from the regular applet file. The name of this applet file is the same as the unsigned applet file except that it is followed by the letter S: for example, MGBC901_03S.cab

Note: You can set your application server to work with a non-persistent applet module by setting the Persistent Browser Client module parameter under the Settings\Environment\Application Server section to No.



Arguments in Asynchronous Events

The Raise event operation can also pass arguments when it is asynchronous (Wait=No). In this case the arguments will always be passed by value and never by reference. The passed arguments will not return changes in the passed arguments values.

MVCS Navigation

When the application is opened in team development, the Navigator has an MVCS view option that lists all of the objects checked out by the user who is currently logged-on.

Cut, Copy, and Paste Magic Operations

The clipboard operations are available on marked lines in the Operations table of a task. This option lets you repeat or move an existing set of operations among different handlers of the same task.

VARPIC Function

The VARPIC function returns a string value that represents the picture of a given field.

For more information about this function, refer to the eDeveloper Reference Guide.

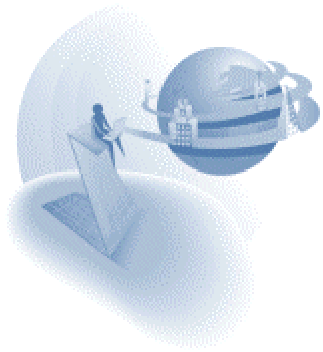
Column Border Properties

The column object of a GUI table now includes two additional properties:

- Top Border
- Right Border

These two properties let you easily define whether the borders of each cell in the table will be painted or not. These properties expect a Yes or No value that will be immediately reflected on the form in toolkit and runtime. These properties also accept an expression for a dynamic value. The expression should evaluate to a Boolean value and is only resolved at runtime. Using an expression allows the borders to be painted for each row in the handled column.

- Top Border – When this property is set to Yes, the top line (row divider above the cell) will be painted. If this property is set to No, the top border will not be painted. If the cell is the top cell of the table, the top border will always be painted.
- Right Border – When this property is set to Yes, the right vertical line (column divider on the right side of the cell) will be painted. If set to No, the right border will not be painted. If the cell is the right-most cell of the table, the right border will always be painted.



Incremental Locate on Magic Selection Lists

The selection lists provided by Magic let you locate an entry by incrementally searching the entry using the name column.

EDITGET Function

This function lets you get the value of a control in its edit mode and is meant to be used from handlers invoked on an editable control.

The EDITGET function returns the edited value of the last control from which the logic, or handler, was invoked.

The EDITGET function also returns the value of the edited control according to the variable's attribute.

In controls other than the edit control, which cannot be edited, the value will be the current value of the control or variable.

For more information about this function, refer to the eDeveloper Reference Guide.

EDITSET Function

This function sets the edited value of the control that invoked the last handler.

The value provided in this function must match the attribute of the variable that is bound to the target control. When the attributes are not the same, the function fails.

For more information about this function, refer to the eDeveloper Reference Guide.

Range on X-ref

A range that lets you restrict the engine to a limited range of objects to search has been added to the cross-reference request form. The range is available for all applicable objects.

New Color and Font Usage

The Magic toolkit screens previously used color entries 1,2 and 5, and font entries 1 and 9, which are default colors and fonts for runtime.

To maintain the separation between toolkit and runtime, the toolkit screens have been modified to use other color and font entries, as listed below:

- The use of font 1 has been changed to font 23.
- The use of font 9 has been changed to font 27.
- The use of color 1 has been changed to color 18.
- The use of color 2 has been changed to color 19.
- The use of color 5 has been changed to color 20.

The fonts file remains as is; the corresponding 23 and 27 entries already exist.

The colors file has been changed. The required entries did not exist before. Entries 18,19, 20 have been created.

If you wish to keep your previous color file, simply change the 18th,, 19th, and 20th entries of your color file to:

Window,FFFFFFF7,FFFFFFFA

Edit Field,FFFFFFF7,FFFFFFFA

3D Effect,FFFFFFED,FFFFFFFA

Registered Splitter Location

Magic keeps the last position of the splitters within the different toolkit screens.

eDeveloper 9.01 Service Pack 2

New Features

Handling of Mouse Events on the Magic Client

You can now easily handle and invoke mouse events, such as Click, Double Click, Mouse Over, and Mouse Out, on the Magic client window.

Last Repository Location

When you return to a repository, the cursor will automatically be parked on the entry in the folder where the cursor was parked the last time that repository was opened.

Jump to Folder

You can easily jump to a different folder in a repository by choosing the Jump to Folder option on the Options menu, or by clicking on CTRL+F to select the folder you want to jump to.

Grid Lines in the Property Sheet

The property sheet now has grid lines for better readability.

Internal Events Added to the Keyboard Mapping File

The internal events listed below have been added to the keyboard-mapping file so that you will be able to define your own shortcut keys for these events:

Check out Object, Check in Object, Resync Objects, Uncheck Object, Horizontal Equal Spacing, Vertical Equal Spacing, Move Forward, Move Backward, Send to Back, Bring to Front, Display Z-order, and Fit to Sizes.

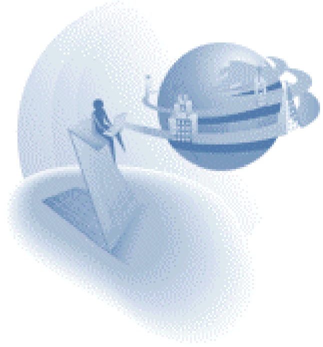
STRTOKENCNT function

This function returns the number of existing delimited tokens in a given string.

For more information about this function, refer to the eDeveloper Reference Guide.

CR in Column Title

The text of the column title displays the line breaks correctly in multi-line format. If you want to write a multi-line text in your column title, you should activate the Wide option (F6) on the column title property in the property sheet



eDeveloper 9.01 Service Pack 3

New Features

New VAR functions

Two new functions have been added to let the Magic developer refer to a variable using its name.

VARCURRN Function

This function returns the current value of a variable according to the variable's name. You can now easily handle and invoke mouse events, such as Click, Double Click, Mouse Over, and Mouse Out, on the Magic client window.

VARINDEX Function

This function returns the index of a variable according to the variable's name.

The VARINDEX function can be used in other VAR functions such as VARMOD, VARATTR, VARPREV, VARSET, VARPIC as the variable index.

For more information about these functions, refer to the eDeveloper Reference Guide.

Setting a Variable's Name in a Select Real Operation

The name of a select real variable can be changed to a name that is different from the name defined for it in the table.

The name column of the Select Real operation can be edited to enter the new name, which will be displayed whenever the variable is used or viewed throughout the program.

This new name is only relevant for the program, and its subtasks, in which it is defined .

You may not enter a blank name. Doing so will change the variable's name back to the name that was defined in its table.

EVALSTR Function

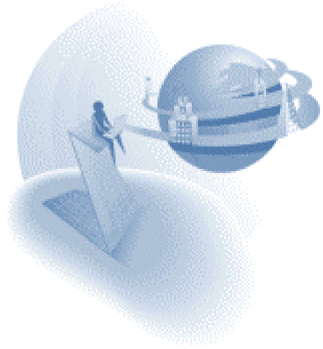
This function lets you evaluate dynamic expressions that may be constructed at runtime.

For more information about these functions, refer to the eDeveloper Reference Guide.

Duplicate Displayed value

A data control may display duplicate values of the displayed field in the choice control. Selecting any one of the duplicate options returns the corresponding linked field.

Duplicate value options of the linked field are disregarded and are not displayed. Only the first occurrence of the linked field value is displayed.



Data Control Caching

The view created by a choice control is cached for every data control.

- ❑ The view is cached according to its defined range criterion. The view of a data control will be taken from the cache only if the full range criterion matches a previously fetched view of the same data control.
- ❑ The cache of the various views is handled separately for each select operation, and it is not reused for identical data controls of other select operations. Each data control of a specific variable has its own cache of records.

New Function and Action lists

The function and action lists have been modified to provide grouping of the items in each list.

SETCRSR Function Enhancement

The SETCRSR function has been updated to support all types of window cursors.

This function expects a numeric parameter that determines the cursor shape. The expected numeric values and the cursor shape they produce are listed below:

4. Standard arrow
5. Hourglass
6. Hand
7. Standard arrow and small hourglass
8. Crosshair
9. Arrow and question mark
10. I-beam
11. Slashed circle
12. four-pointer arrow pointing north, south, east, and west.
13. Double-pointed arrow pointing northeast and southwest
14. Double-pointed arrow pointing north and south
15. Double-pointed arrow pointing northwest and southeast
16. Double-pointed arrow pointing west and east
17. Vertical Arrow.

Requester Function Modifications and Additions

The RQRTINF function has been modified, and the functions listed below have been added to support the context and threads related to the handling of Magic Request Engines.

RQRTINF – Modified Function

The retuned value of this function has been modified to provide more information about thread capacity and usage of the given engine. At the end of the information string, the following information is provided:

- Running threads: The number of running threads.
- Threads peak: The maximum number of open threads that was reached.
- Allowed threads: The maximum number of allowed threads.
- If the engine setting is zero, then it should return the maximum allowed according to the given license.

RQRTCTXS – New Function

This function loads the information of all of the opened contexts of the application server in which it is evaluated. The function returns the total number of these contexts. The RQRTCTXS function needs to be evaluated prior to the RQRTCTX function that queries a given context.

RQRTCTX – New Function

This function returns the information of a given context entry of a given service or server name. The RQRTCTXS function should be run before using this function.

RQCTXTRM – New Function

This function terminates a given context. The context is identified by its entry number as retrieved by the RQRTCTXS function.

RQCTXINF – New Function

This function returns an information string of a given context identified by the context ID.

For more information about these functions, refer to the eDeveloper Reference Guide.

HANDLEDCTRL function

This function returns the name of the control from which the current handler has been invoked.

For more information about this function, refer to the eDeveloper Reference Guide.

Browser Client Error Messages Window

A new parameter passed to a browser client instructs the browser client to check for validity of the HTML page and, in a separate window, informs the user about the errors found in the parsing stage. The parameter to be passed is DEBUG_CLIENT=1.

When this parameter is provided, a window is opened for every browser window that lists the errors found in the parsing stage.

When this parameter is not provided or set with a value other than 1, the browser client does not check for the HTML page validity and does not inform the user about the parsing errors.

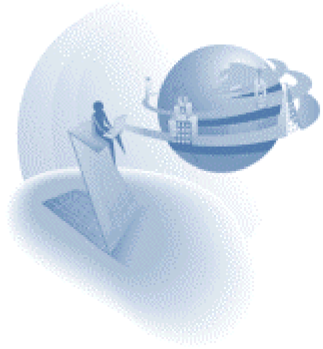
When a browser task is run by pressing F7, the parameter DEBUG_CLIENT=1 will always be added. This means that the validity checks and the errors window will always be displayed for a browser task that is run from the toolkit.

An example of a URL that invokes the error window is:

```
http://host/mgrqispi9.dll?APPNAME=APP&PRGNAME=MYPROG&DEBUG_CLIENT=1
```

Combined Navigator and Property Sheet Windows

The Navigator and Property sheet can be displayed in the same single window. If you want to have them both displayed in the same window, drag one of them onto the other. Then a two-options tab will be displayed at the bottom of the window, allowing switching between the Property Sheet and the Navigator. If you wish to separate them, simply drag the title of the current option while pressing the CTRL key. The combined window can either be docked to one of the Magic window borders or kept as a floating window.



MAIL Functions

A new set of functions has been added to support email functionality. These functions support SMTP for sending mail, and POP3 and IMAP for receiving mail.

MAILCONNECT

This function opens a connection to a mail server. The SMTP server is used to send mail, and does not require user and password. When this function is used to connect to a POP3 or an IMAP server, Magic will connect to the user's mailbox on the server. Upon connection, the MAILCONNECT function retrieves the new mails in the user's mailbox. New mail that the user receives after the connection has been made will not be available during that connection session but only after a new connection is made. Only one connection can be set to a send mail server and one connection to a receive mail server. If connected twice, the second connection overrides the first.

MAILDISCONNECT

The MAILDDISCONNECT function closes a connection to an email server.

MAILSEND

The MAILSEND function is used to send an email. It requires all the information needed to send an email.

MAILMSGID

This function returns the unique id of an email message.

MAILMSGFROM

The MAILMSGFROM function returns the address from which the message was sent.

MAILMSGTO

The MAILMSGTO function returns a comma-delimited string of all the main addresses to which the message is sent.

MAILMSGCC

The MAILMSGCC function returns a comma-delimited string of all the CC addresses.

MAILMSGSUBJ

The MAILMSGSUBJ function returns the subject string of the message.

MAILMSGTEXT

The MAILMSGTEXT function returns the body text string of the message.

MAILMSGFILES

The MAILMSGFILES function returns the number of attachments of the message.

MAILMSGFILE

The MAILMSGFILE function returns the file name of the specific attachment of the

MAILFILESAVE

The MAILFILESAVE function lets you save a message attachment to a file on a disk.

MAILMSGDEL

The MAILMSGDEL function deletes a message from the server mailbox.

MAILERR

The MAILERR function translates a given mail error code that has been returned from one of the functions described above to a readable error message.

For more information about these functions, refer to the eDeveloper Reference Guide.

HTTPGET

The HTTPGET function lets you retrieve the returned HTML result of an HTTP request as a blob.

For more information about this function, refer to the eDeveloper Reference Guide.

Find and Replace Magic Objects

In Toolkit mode a new option has been added that lets you automatically find and replace any Magic object with another object of the same type. All you need to do is to park on the object you wish to replace and select Find and Replace from the Options menu

System Logon Environment Setting

The System Logon setting is located at Settings/Environment/System.

This setting determines the level of the Magic environment integration with the operating system. The valid values for this setting are:

None – The Logon dialog may be opened either automatically, according to the Input Password setting, or explicitly, by activating the Logon menu option without using any default values as the user name.

User Name – The Logon dialog may be opened either automatically, according to the Input Password setting, or explicitly, by activating the Logon menu option, and the default value for the Logon name will be the current user who is logged on to the operating system.

Full Logon – The Logon dialog will not be available, the Input Password setting will be ignored, and the Logon menu option will be hidden. Magic will automatically log on the operating system's current user.

Note: The user password in the Magic user file does not have to correspond to the operating system password. When set to Full Logon, Magic does not check the Magic user password because the user has already been identified by the operating system.

We recommend that even if you use the full Logon option, you should keep a unique Magic password for each user to prevent users from trying to logon as different users.

When the setting is set to Full Logon, if the logon procedure fails, because the system user name does not exist in the Magic users file, the Input password environment parameter will not be ignored. If the Input password is set to Yes, it will open the Logon dialog with the user name of the user who is logged on to the operating system.

IO Device Open Timing Environment Setting

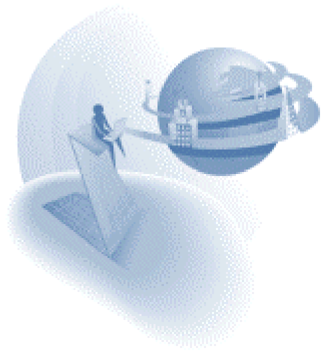
The IO Device Open Timing setting is located at Settings/Environment/Preferences.

The timing of opening an IO device can now be controlled in two ways: immediately as the task that defines it is opened, or on Demand, when the first output or input operation to the IO device occurs or when an IO-device-related function, such as EOF, EOP, LINE, or PAGE is used. The default option is Immediate.

Valid values: Immediate, On Demand

Immediate – When set to Immediate, any IO device will be opened when the task that defines the IO is opened.

On Demand – When set to On Demand, an IO device will be opened at the first execution of an output or an input operation that is set for the IO or for the evaluation of an IO-device-related function, such as EOF, EOP, LINE, or PAGE.



eDeveloper 9.2

New Features and Modifications

J2EE Integration - Release

Using Magic eDeveloper V9.2 you can now generate EJBs in Magic for use within a J2EE environment. For more information, refer to Chapter 23: J2EE Integration of the eDeveloper Reference Guide.

Variable Scope in Event Handlers

The variable stack available to a program in runtime, including all the current task variables and all the tasks above it in the runtime tree, is now also available for a handler defined in an ancestor task.

Remote Flow Monitor

You can now monitor the flow of any Magic application from a remote client. For more information please refer to the eDeveloper Reference Guide.

Team Development with Components

Magic eDeveloper enables team development of an application containing Magic components that are also in a team development state.

To allow this you should place the following keyword in your Magic.ini file under the [MAGIC_SPECIALS] section:

TeamDevelopmentForComponents = Y.

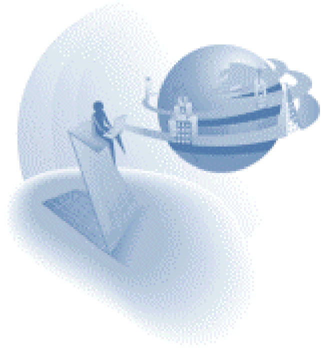
Adjustable Variable Palette

You can now freely adjust a Form Editor's Variable Palette. The font entry number 37 allows you to set the font you prefer the Variable Palette to use.

Text Find & Replace

Magic eDeveloper Version 9.2 lets you find any occurrence of static text used throughout the application. You can also use the Replace option to replace existing text with new text.

These options are available from the Edit menu.



Requester Upload Capability

Magic eDeveloper Version 9.2's Internet requester provides new upload functionality from your HTML page. You can enable a file to be uploaded from the HTML page and then have it received by a Magic program.

To enable this you should define your HTML file as follows:

1. The form method attribute type must be POST. for example:
   `<form action= /magic92scripts/mgrqispi92.dll method=post>`
2. The form tag should include the following attribute: ENCTYPE="multipart/form-data", for example:
   ```
   <form action= /magic92scripts/mgrqispi92.dll method=post
   ENCTYPE="multipart/form-data">
   ```
3. You should add to the form an Input tag of type: FILE. This tag creates both an edit box in which the user can enter the local path and file name, and a push button allowing the user to browse and select the file. This input tag should be identified by a clearly defined name. for example:
   `<input type=file name="MyFile">`
4. The given name of the file input tag should be added to the hidden input of the arguments. for example:
   ```
   <input type=hidden name="ARGUMENTS"
   value="var1,var2,....,MyFile,...varn">
   ```
5. A corresponding BLOB virtual variable should be created in the called Magic program. This variable is updated with the uploaded file content.

The variable's content can be converted to a physical file on the server side using the BLB2FILE function.

A sample HTML file (Upload.html) can be found together with the HTML information on the eDeveloper V9.2 CD.

Additional Mail Functions

Three new mail functions have been added:

MAILMSGDATE

This function retrieves the date and time information of the given mail message that is identified by its index. The date and time information is returned as a string

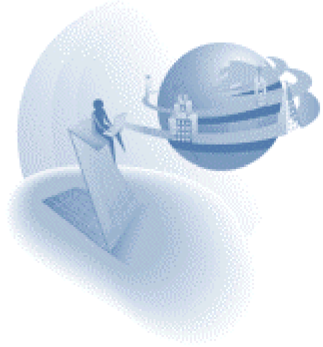
MAILMSGREPLYTO

This function retrieves the “reply to” string of the given mail message.

MAILMSGHEADER

This function retrieves the header information of a given mail message.

This function may retrieve the entire header information or the value of a given header-key.



Default Date value and Default NULL Display Strings

You can define default NULL display strings and a default date value for the entire application.

These values are defined in a new section of the Magic.ini file: [MAGIC_DEFAULTS]

Note: You can only define these settings by editing the Magic.ini file. The settings are not available in Magic's Environment dialog box.

These settings are regarded as the system default values for the default value of a date field definition and the NULL display string for all other field attributes.

If these settings do not exist in the Magic.ini file, Magic uses its own already defined defaults.

The following is an example of the section and its available entries in the Magic.ini file:

```
[MAGIC_DEFAULTS]
DefaultDate = [date value]
NullAlphaDisplay = [string value]
NullNumericDisplay = [string value]
NullLogicalDisplay = [string value]
NullDateDisplay = [string value]
NullTimeDisplay = [string value]
NullMemoDisplay = [string value]
NullBlobDisplay = [string value]
```

Note:

6. The date format should adhere to the defined date mode.
7. Any modification to these settings during the execution of an application is effective upon the next session of Magic

Browser Client Language Support

A new Environment setting has been added allowing you to define the character set required for display and data input on your HTML pages.

Browser Task ANSI to Unicode Translation

Use this setting to define the name of the encoding table to be used by the HTML page of a browser task. When this setting has a defined value, each HTML page created by a browser task instructs the page to use the defined character set by embedding a META tag:

```
<META HTTP-EQUIV="Content-Type" content="text/html; charset=XXX">
```

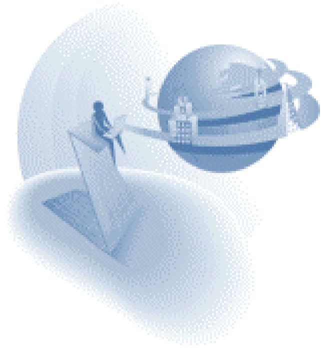
Where XXX is the string taken from the Environment setting.

This information is also used to instruct the browser client module to use the defined encoding table for translating the entered data that has been passed to the Magic application server engine.

Note:

8. If your HTML page already has a META tag, this tag remains unchanged. However, the browser client module will still be instructed to use the encoding table defined by the Environment setting.
9. The encoding table defined in this setting is case sensitive. You may refer to these links that list the supported encoding tables and their codes:
   http://java.sun.com/products/jdk/1.1/docs/guide/intl/encoding.doc.html
   http://www.devsphere.com/mapping/docs/guide/encodings.html

For more information about this setting, refer to the eDeveloper Reference Guide.



eDeveloper 9.2 Service Pack 1

Important Notes

Pervasive. SQL™ 2000i Gateway 9.2-9.0

Service Pack 1 provides a new gateway (MGp2k.dll) for accessing the Pervasive.SQL ™ 2000i database. Both the desktop and the enterprise products support this gateway. The Pervasive gateway lets you connect to a Pervasive engine using the following two methods:

1. Defining a DSN (data source name) that points to the engine being used, be it local or remote. The DSN name has to be specified in the DBname column within the [MAGIC_DATABASES] section entry of the Magic.ini file.
2. For remote access to a Pervasive engine, you can use the DSNless connection by specifying:
   a. The server machine name in the 'Database Server' column within the Database properties of the database entry in the [MAGIC_DATABASES] section of the Magic.ini file.
   b. The remote DSN name (as defined once on the server machine) in the DBname column within the [MAGIC_DATABASES] section entry of the Magic.ini file.
   c. Note that if you have some kind of Pervasive engine installed on the client machine, be sure to define your Target Engine as "Try Server then Workstation" or "Server only" in the client configuration, Pervasive.SQL control center

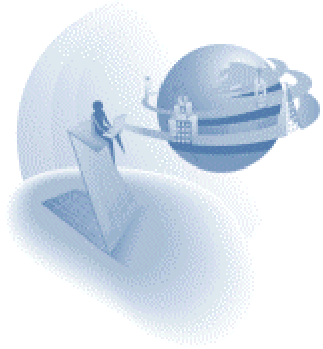
RIGHTS Function in EVALSTR Function

The RIGHTS function used in the EVALSTR function always returns a True value whether the logged-on user has the specified right or not.

Please refrain from using the RIGHTS function within the EVALSTR function.

How To… Help

The Magic eDeveloper *How To…* manual offers a walkthrough guide for utilizing the various capabilities of the tool. This manual is now provided as a Help file. You can access the *How To…* Help by clicking Help Topics on the Magic eDeveloper Help menu and selecting the *How To…* section in the Contents.



The CISAM Gateway and the Magic Data Server

As of Version 9, MSE has ceased development of the CISAM gateway and the Magic data server. The items below cover the different aspects and uses of the CISAM gateway and the Magic data server.

Accessing CISAM data

Magic Version 8 only allowed access from the Magic client to CISAM files through the Magic data server using the CISAM gateway that resided on the UNIX server machine.

Magic eDeveloper 9.2 Service Pack 1 lets you access CISAM data files using the Magic gateway for Informix.

In order to migrate your Version 8 application that accesses CISAM data to eDeveloper, please refer to the document and conversion application at the following links: http://v9product:PrEver983@ftp.magicsoftware.com/v9product/CISAMAccess/ http://v9product:PrEver983@ftp1.magicsoftware.com/v9product/CISAMAccess/

The same capability of accessing CISAM data files for the Magic application servers is available using the Informix gateways on the different UNIX platforms starting from Version 9.2.

Note that no data conversion is necessary. The only modification to your application is within the Table repository.

Accessing a Magic Control File (MCF) stored in CISAM

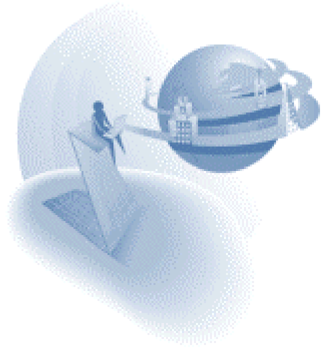
Magic Version 8 let you store the MCF in a CISAM file. This option was used in both the development and deployment phases. Magic eDeveloper lets you store your MCF during the development phase in any of the supported databases. There is no need to use any of these databases in the deployment phase because you can save your MCF as a Magic Flat File (MFF). The MFF is a binary flat file that is platform and database independent. You can copy the MCF from platform to platform, in binary mode, and the MCF can be accessed by the corresponding Magic Application Server.

Accessing databases using the data server

The Magic Version 8 data server let you access RDBMS data using the various gateways residing on the server machine. Magic eDeveloper provides access to these databases during both the development and deployment phases using the database client software. In addition, during the deployment phase, the Magic Application Server can access the RDBMS data using the various gateways that reside on the different server platforms.

Additional data server functionality

Other data server functionality, such as executing commands on the server, printing to server printers, and direct access to files on the server, can be achieved using the Call Remote command or other third party tools.



New Features

Limitation of File Upload

A new setting, `MaxUploadKB=nnnn`, has been added to the Mgreq.ini file. This setting determines the maximum size allowed for a file upload. The size is defined by KB bytes. If a file larger than the limit is submitted, the requester will fail the request and return an error back to the client. The error message text is: "File upload failed: file size exceeds maximum size limitation." The Error code is: -260 If this parameter is not defined, is set to zero, or is a negative value, Magic does not limit the size of a file to be uploaded. Unlimited size of uploaded files can create congestion on the Application Server.

RUNMODE Function Modification

The RUNMODE function has been modified to reflect a new state, running the application for the first time in an Application Server. When the RUNMODE function is evaluated, the main program returns the value of (-1) if the application has been opened for the first time on a background Application Server. The RUNMODE function returns 0 in the following cases:

For more information about this function, refer to the eDeveloper reference guide.

HTTPPOST Function

The HTTPPOST function lets you post information via an HTTP message and receive an HTML\XML result of the HTTP request back as a blob.

For more information about this function, refer to the eDeveloper reference guide.

Clipboard Functions

A series of functions has been added to enable interaction with the clipboard. Writing to the clipboard can be done by creating data to the clipboard (CLIPADD function), and then by writing to the clipboard (CLIPWRITE function). The CLIPREAD function lets you retrieve the content of the clipboard

CLIPADD

This function receives a number of parameter pairs. Each pair contains a value and the picture to use when placing the value on the clipboard.

CLIPWRITE

The CLIPWRITE function returns true if the buffer was successfully placed in the clipboard.

CLIPREAD

The CLIPREAD function returns the contents of the clipboard.

For more information about these functions, refer to the eDeveloper reference guide.

Floating Window - Initial location

In runtime a floating window always opens centered, regardless of its top\left settings.

A special flag lets you change the way floating windows are opened:

`[MAGIC_SPECIALS]`
`SpecialCenteredFloatingWindow=Y\N` (default: Yes).

When this flag is set to No, any floating window will be opened as according to the way its Top\Left properties have been defined.

When this flag is set to Yes, any floating window will be opened centered.

Note: A modal window is also a floating window.

Change Effective: immediate.

Dynamic Spool Name for Graphical Printers

When using a logical name as the I/O name in the I/O files list, Magic names the spooled file as "Magic - " concatenated with the interpretation of the logical name. A nonexistent logical name results in an empty interpretation. This feature is mainly meant for usage with the PDF driver, thus enabling dynamic PDF file names.

Event Handling – Handler's Scope Modifications

The scope of resources available for a handler has been modified to match the scope of resources available for a Version 8 event.

Querying the runtime tree

Up until Version 9.3, a handler of an ancestor task that was invoked by a descendant program could query the application using the generation-related functions only from the program that keeps handler an upwards. Now such a handler can query the full runtime tree.

For example, Program A called Program B (having a handler for 'ABC' event), which in turn called Program C. Program C raised the 'ABC' event. Before Version 9.3, the handler could only query through Program B (generation=0) and Program A (generation=1). In Version 9.3, the handler can query through the entire runtime tree; i.e., Program C (generation=0), Program B (generation=1), Program A (generation=2).

This modification is available both in Client/Server applications and browser-based applications.

Browser Client task - Exception

Note that if a handler of a browser task called another browser task in a modeless window, then the runtime tree of the called task will be only from that called task through the task in which the handler is defined.

IO

A handler invoked in a task and defined in an ancestor task can output or input forms to an IO defined in the full runtime tree using the 'IO name to use' IO property.

You may download a sample application that demonstrates this new feature.

Use this link: Handler_Scope.zip

New Features and Modifications

Sample applications

Some of the following new features provide a sample application that demonstrates the new feature or new behavior. A link to the sample application zip file is noted in each feature that has a sample application. These zip file contains the required files for the sample application and a readme file that will guide through.

Note:

Some file names of the various samples are the same, so we suggest you extract the files from each zip file to a different directory.

Web Services Support

The Web Services Consumer support is included with this version of Magic eDeveloper. Please refer to the Web Service Consumer.htm file for more information.

The Web Services Provider support is included as a beta version.

Please refer to the Web Service Provider.htm file for more information.

Scripts directory

The files used for Internet support are now placed separately in a sub-directory named *Scripts* under the Magic installation directory. The *Magic93Scripts* web server alias points to the new *Scripts* sub-directory.

Browser Client – Overriding a browser task

The browser client closes a child task that is replaced by another task properly using a call operation.

You may download a sample application that demonstrates this new behavior Use this link: Overriding_A_Browser_Task.zip

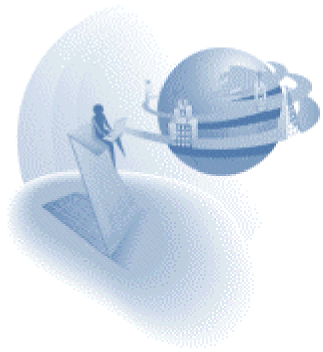
Browser Client – Overriding the top browser task

The browser client properly closes the top browser task, the first browser task to be run, when it is replaced by a call operation invoked from the main program.

Note: The destination of such a call operation should be set to _top.

You may download a sample application that demonstrates this new behavior

Use this link: Overriding_A_Top_Browser_Task.zip



Browser Client – Keyboard access to subforms

A subform in a browser client task can be accessed from the keyboard, using Tab\Shift+tab keys, and not only by the mouse. Subforms are integrated with the parent task tab order by placing the subform in the HTML controls list as follows:

- ❑ The first parkable control of the subform will be next to the parkable control of the parent task that is defined before the subform in the HTML controls list (‘Control A’).
- ❑ Tabbing from the last parked control of the subform moves the focus to the parkable control of the parent task that is next to ‘Control A’, according to the select operations order of the parent.

Tab into – New property

The subform control provides a new property that defines whether the subform should be accessed by tabbing or not. If the property is set to Yes then the subform will be accessed by tab according to the rules mentioned. If the property is set to No, then the subform will not be accessed by tab, only by mouse click. The default value of this property in a newly created subform is Yes. The value of this property in subforms that were defined in a previous version will be set to No.

You may download a sample application that demonstrates this new behavior. Use this link: Keyboard_Access_Subforms.zip

Browser Client – Flow monitor support

The flow monitor displays the flow of a browser client task when it is executed on the server side and on the client side.

Browser Client – XML encryption

The XML portion of a browser client task in the source of the page is encrypted. This way the logic of the task and its initial data cannot be browsed.

Browser Client – Style & Class support

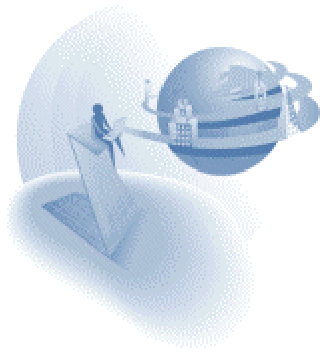
The browser client tasks let you dynamically set the attribute of an HTML control by assigning it either a class name or a set of styles.

Each HTML control that appears in the HTML controls list has the following new properties:

- **Default Class** – This property describes the class by which the HTML control will be displayed. The value of this property is set by an expression. Once the expression will be re-computed, the class of the HTML control will be set by the new class value. The expression is expected to return a string value that represents a class name defined in the HTML page.
- **Mouseover Class** – This property describes the class by which the HTML control is displayed when the mouse cursor is placed on top of the control. The value of this property is set like the default class. When the mouse cursor moves off the control, the class returns to being defined by the 'Default class' property.
- **Mousedown Class** – This property describes the class by which the HTML control is displayed when the left mouse button is pressed while the mouse cursor is placed on top of the control. The value of this property is set like the default class.
  As the left mouse button is released, the class returns to being defined by the 'Mouseover class' property when the cursor remains on top of the control or, as defined by the 'Default class' property when the button is released after moving off the control.
- **Default Style** – The Default Style property lets you define a list of styles that sets the attributes of the HTML control. You need to define the attributes for each style, its name, and its value. The value is set by an expression. The expression is expected to return a string value that represents the style's value. When the expression is re-computed, the style of the HTML control is defined by the new value.
- **Mouseover Style** – The Mouseover Style property sets the style values by which the HTML control is displayed when the mouse cursor is placed on top of the control. The values of the styles are set like the Default style.
  When the mouse cursor moves off the control, the class returns to being defined by the Default style property.
- **Mousedown Style** – The Mousedown Style property sets the style values by which the HTML control is displayed when the mouse button is pressed while the mouse cursor is placed on top of the control. The values of the styles are like the Default style.
  When the left mouse button is released, the styles return to being defined by the Mouseover Style property when the cursor remains on top of the control, or by the Default style property when the button is released after moving off the control.

Note that the names of the styles are the same as used in Java script and HTML syntax.

You may download a sample application that demonstrates this new feature using this link: Styles_and_Classes.zip



MMCLEAR – New Function

The MMCLEAR function clears the marked records in a table control.

For more information on this function, refer to the eDeveloper Reference Guide.

MNUNAME – New Function

The MNUNAME function sets the menu entry text of a given menu entry.

For more information on this function, refer to the eDeveloper Reference Guide.

Team Development Modifications

The Team Development capability of Magic eDeveloper has been enhanced in Version 9.3 to provide better handling of shared applications and faster team development response time.

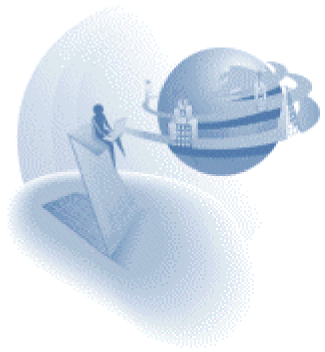
Modifications to the Program Repository

The Program repository maintains the application’s Program list order. Every Program list modification requires that you update the Program repository.

In previous Magic versions, the Program repository was automatically checked out and checked in for every Program list change. In Magic eDeveloper Version 9.3, the Program repository is not automatically checked out and checked in for modifications to the Program list. The developer must manually check the Program repository out and in.

You must check out the Program repository to modify the Program list. The Program list can be modified by:

1. Creating a new program
2. Deleting a program
3. Changing the sequence of programs
4. Moving a program to another program folder
5. Renaming a program folder
6. Overwriting a program
7. Repeating programs
8. Assigning rights to the program repository
9. Importing a program
10. Generating a program from the Table repository



Checking the Program repository out and in

You can check the Program repository out or in by clicking the Check Out or Check In options on the MVCS menu, or by clicking the Check Out or Check In toolbar buttons. You cannot uncheck the Program repository.

Importing Programs

When you import an application that has programs, you must check out the Program repository. When importing a Magic eDeveloper V9.3 export file, Magic prompts you to check out the Program repository. If you do not check out the Program repository, or if the Program repository has already been checked out by another developer, Magic issues a warning that the import process cannot be performed.

When you import an application from a previous Magic version, the import process cannot detect the programs included. You will only be prompted when Magic, through the import process, finds a program. If you do not check out the Program repository or if the Program repository has been checked out by another developer, Magic issues a warning that the import process cannot be performed. The import process will fail and the application will be rolled back.

Argument Matching

Due to team development modifications, the Argument list of a Call operation can only display a maximum of 255 expected arguments.

Table Control – Last Divider – New Property

The last divider property determines whether the last divider of the table control, mainly its anchor on the title bar, is displayed or not. If you set this property to No, the last divider of the table will not be displayed.

Magic Broker – Load Balancing Enhancements

The Magic broker load balancing mechanism has been improved to provide better load balancing between the available processes. The mechanism has been improved so that requests will be directed to the engine that has the best performance. If more than one engine has the best performance, requests will be directed to the engine among those of equally good performance that keeps the least number of active contexts.

Magic toolkit palettes – Switch panes

The toolkit action of switching panes (CTRL+TAB) has been modified to switch between all opened palette windows and not just between the workspace and the navigator.

Input/Output – New properties for Right-to-Left (RTL) applications

Two new properties have been added to the IO definitions of File and Printer. These new properties are relevant for Right-to-Left (RTL) applications.

Visual to Logical

The Visual to Logical property controls the visual to logical conversions of the processed data. When this property is set to Yes, in the output process each field is converted from logical to visual and the entire line is converted from visual to logical.

When this flag is set to No, no conversion of visual to logical or vice versa will be performed.

Flip Line

The Flip Line property controls the flip of the entire line when it is exported or imported. When set to Yes, the line is flipped. When set to No, the line is written or read as is.

This property is relevant only when the Visual to Logical property is set to Yes or evaluates to a True value.

When the Visual to Logical property is set to No, this property is disabled.

The default value of this property is No.

Magic Toolkit Palettes – New Settings

Three new environment settings enable you to customize the behavior of the toolkit palettes:

- ❑ Floating palettes always on top
- ❑ Dockable palettes
- ❑ Single expand palettes

For more information on these environment settings, refer to the eDeveloper Reference Guide.

Security Mechanism – Directory Service Support

Magic lets you set user security settings using directory services. Magic reads all groups that a user belongs to and matches these user groups with groups defined in the standard Magic security file. If a match is found, all the rights that belong to that group will be granted to the user.

It is important to note that under this schema, there is no need to define individual users in the Magic security file, only groups.

Setting the environment parameter of ‘Logon’ to ‘Directory’ activates this option. Once set to ‘Directory’, the logon to the Magic engine is performed automatically according to the user who is currently logged on to the network.

A new environment setting lets you define a binding string that is required for Magic to work with the directory service:

Directory Binding String

Setting/Environment/External

The Directory Binding string is used to bind into the directory service. During the binding process, Magic automatically concatenates the operating system username to the string and attempts to bind into the operating system.

The syntax for Win32 binding is:

```
WinNT:[//DomainName/[ComputerName/[ObjectName[,className]]]]
```

Due to the fact that not all operating systems have active directory support installed as a default, a new DLL file is provided to support active directory by Magic . The DLL file, Mgactdir.dll, is installed during the installation process.

External Event – Invoking the Magic process externally

A new event has been added called "External Event". This event enables easier integration between Magic and external programs running on the same computer, by enabling Magic to receive and process events and data from external programs.

Using Windows API

Raising an external event in Magic is done using a SendMessage window API command with the WM_COPYDATA message type. This message contains a pointer to a COPYDATASTRUCT structure. Magic places these parameters into the event queue, and the handler receiving the event receives its parameters.

COPYDATASTRUCT Format - This structure contains the three members listed below:

- *dwData* – number indicating the number of parameters sent
- *lpData* – pointer to a buffer containing the parameters
- *cbData* – length of data in lpData.

Parameters - The parameters placed in the cbData member are placed as null terminated strings in the buffer. Magic accepts four types of parameters:

- -A – Alpha parameter
- -N – Whole numeric parameter
- -F – Floating point parameter
- -L – Logical parameter (T or F values)

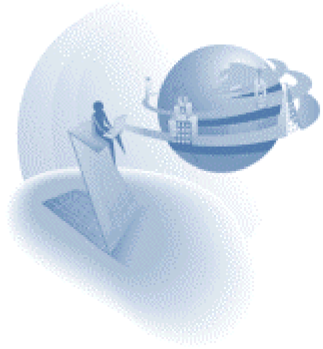
Example: Suppose we are to send 4 parameters containing "XXX", 17, True, 643.23. The created buffer should be in the following form:

```
-AXXX☺-N17☺-LT☺-F643.23☺
```

Where the ☺ sign, represents a null character.

Defining the handler

To handle the external event, you need to create a handler on the 'External event' part of the Internal type of events. In this handler you can create the virtual variables that will receive the passed arguments.



Demo executable and CPP code

The Magic installation provides an executable file, Msg2mgic.exe, that demonstrates the invocation of the external event in Magic.

Usage: `MSG2MGIC [T/R] -[A/N/F/L]=value`

T/R - Send a message to the [T]oolkit or the [R]untime version of Magic

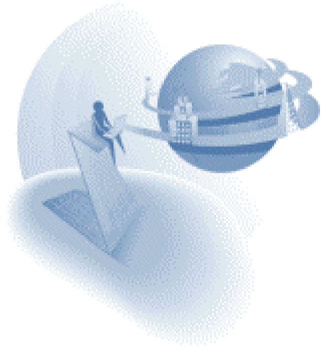
Any number of the following parameters may be sent:

- ❑ [A]lpha variables
- ❑ [N]umeric variables
- ❑ [F]loating point variables
- ❑ [L]ogical values using F or T, for FALSE and TRUE

Example: MSG2MGIC R –ABill –N123 -LT

Note that this example does not let you invoke the external event in an engine run with a demo license.

You can find the CPP source files of the Msg2mgic executable file at msg2magic.zip



eDeveloper 9.3 Service Pack 1

New Feature

Compression of the Browser Client XML Portion

The entire browser task logic is transmitted to the browser in an XML format that is part of the HTML result page of the browser task. The size of this portion becomes larger as the task becomes larger, which results in a large HTML file that needs to be transmitted back to the browser.

Since the release of V9.3 SP1, the Magic Application Server can compress the XML to a smaller size, thereby creating smaller HTML files. When the XML is compressed, the browser client module is required to decompress it. This procedure causes a longer initialization time when the browser task is loaded on the browser.

To define the required tradeoff, of smaller files as against short initialization time, a new Magic Specials flag lets you define the compression size you wish to gain:

```
[MAGIC_SPECIALS]
CompressBrowserXML = nn
```

For example: If you set this flag to 30 Magic will compress the XML only if the compression will make a reduction of more than 30 Kilobytes. In this way, small HTML files are not compressed and their initialization is not affected. Larger HTML files are compressed to provide quicker transmission of the files, but the initialization of these pages is only affected slightly.

eDeveloper 9.3 Service Pack 2

Important Notes

Web Services Provider - Release

The Web services provider feature has been released for deployment. For more information on Web Services support see the eDeveloper Reference Guide.

Supporting the JBOSS Platform for EJB

Magic eDeveloper lets you produce EJB components for a Magic application and for the JBOSS platform. When EJBs are created by Magic's Component Builder an additional directory is created for the JBOSS platform. You can find the files required for deploying a Magic EJB in this directory.

Behavior Modifications

Team Development – Handling requests in toolkit

Every request sent to a Magic engine logs on according to the following rules:

If the request carries details of user name and password the request will log on according to the user information, otherwise the user name and password defined in the requester INI file are used for the logon procedure. If the no user details are set the request will be handled with no user logged on.

When working in team development environment this is significant since you would expect that requests will work on the objects of the application that were checked out and modified by your user.

A request that carries no user name to log on to the Magic engine is executed with no user logged on, hence the request cannot access the objects checked out by your user. To have a request log on to the Magic engine by your user you should define your user name in the requester INI file (mgreq.ini) both in the Magic working directory and the directory of the requester.

```
Username = [Your user name]

Password = [Your password]
```

Browser Client - First Record Prefix

The first Record Prefix handler of the first record, which is executed at the beginning of the task, used to be executed on the server side just after the Task Prefix handler. Other Record Prefixes were executed on the client side. In eDeveloper 9.3 SP2 the first Record Prefix is now executed on the client side when the page is displayed and becomes active. In this way the execution for all Record Prefix handlers is the same.

Multi-marking - Asynchronous Events

Asynchronous events are ignored if they are triggered while the Magic task is processing the marked records. This means that if during the multi-marking handling process an event is raised asynchronously (For example: a Raise event operation with Wait=No, triggered by a key combination, a timer event, or an expression event) this event is ignored and will not be handled either during the process or after it is completed.

However, if during the multi-marking handling process the handler calls another task, such as online, batch or browser, then any asynchronous event raised within the called task will be handled as usual. If an asynchronous event was raised in such a task and remained in the event queue, when the engine returns to the multi-marking process the event will be ignored.

Multi-marking - Reposition After Modify

When the Reposition after modify environment setting is set to work (=Yes), and records are modified during the multi-marking handling process, the reposition will only occur at the end of the multi-marking process. In this case the record marking will be cleared.

Cross-reference Results

Newer cross-reference results are entered at the top of the X-ref result list in the Navigator.

Loading Recursive Components

Magic does not allow the loading of a component that refers to its container application, or if one of the component's nested components refers to the container application. For example: Application A cannot load and use Component B if Component B's application has Application A loaded as its nested component.

The following message is displayed when an application that has this kind of recursive reference is loaded:

```
Application file cannot be opened due to recursive components.
```

The components that failed to load due to this constraint are not accessible for the container application both in Toolkit and at Runtime, as though the component's application file cannot be found.

Field Model Selection List

The field model selection list has been modified and displays two additional columns:

- ❑ Attribute - Displays the field’s attribute.
- ❑ Picture - Displays the field’s picture.

Data Control Range Expression Defined on a Local Variable

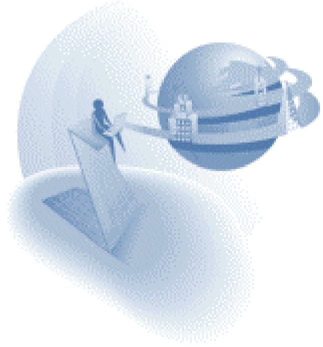
The ability provided in V9.20 SP1 to define a data control's range expression from the local variable has been removed.

If you import a program in which local variables are defined in this manner, the defined range expressions will be set directly on the controls that refer to the variable. This is will be done only when the controls are of the same task and the range property of the controls is inherited.

The same modification occurs when you enter this kind of program without importing.

Closing the Main Program at Runtime, Within Toolkit

When the main program is closed in toolkit mode while being run at runtime, the application will not be closed. Instead the application will switch back to toolkit. This is relevant when you initially open the application in runtime, or when you switch to runtime by pressing CTRL+T or when you press F7 to run the APG.



X-Ref, Find & Replace, Find Text, and Replace Text

The X-Ref, Find & Replace, Find Text, and Replace Text utilities look for the specified object or text only in broken properties. Properties that inherit the searched text or the reference to the searched object will not be included in the search results.

Raise Event - Default Wait Parameter

When a new type of event is selected in the event selection dialog from and an expression is not set for the wait parameter, Magic automatically sets the value as follows:

For an internal event type the Wait property is set to 'No', for other event types Wait is set to 'Yes'.

Browser Client - Persistent Applet

The Persistent applet of the browser client has been improved to suit the security scheme of the browser. With the new persistent applet the Launcher.cab file used to load the persistent applet is no longer required for providing applet persistency. Only the signed applet is required. As a result the Launcher.cab file is not provided with the installation. There is now only one confirmation dialog for the signed applet. If you confirm the signed applet using the Yes option the dialog will not open even after you close and re-open the browser, until the next version of the applet.

Bottom Placement of Table Control in Create Mode

The table control is affected by bottom placement when you add or remove table rows in Create mode.

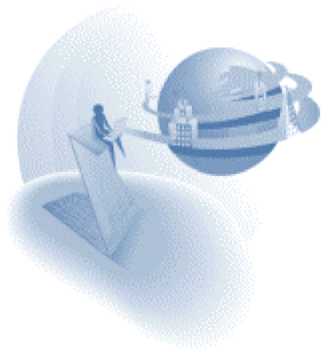
Multi-marking with Scrollbar

Using the scrollbar for navigation while pressing CTRL preserves record marking.

Removing the 'Record is no longer in view' Message

The ‘Record is no longer in view’ message is no longer displayed when you perform the View Refresh event, and the last parked record cannot be parked on.

Note: The message still appears if you cannot park on the previous record after Screen Refresh and Record Flush.



Event Selection Dialog

The behavior of the Event selection dialog has been modified to provide a faster selection of events.

The event selection dialog is available in the following locations:

1. Handler level
2. Raise event operation
3. Push button control – Raise event property
4. Menu – Event menu type

Default Event Type

When the event dialog is opened for the first time, within the current toolkit session, the Event type will be ‘User’.

Keep Last Used Event Type

When you open the Event selection dialog again the event type will be the last selected event type in the current toolkit session.

Automatic Zoom

When you move to the Event field and the field is empty Magic will automatically zoom, if zoom is applicable, and do the following:

1. System: will open the Key Definition dialog
2. Internal: will open the Action list dialog
3. User: will open the Event list dialog.
4. Timer: there will be no zoom.
5. Expression: will open the Expression editor.
6. Error: will open the Error list dialog

Soap Server Definition

When the Magic engine is instructed to load the SOAP gateway, through the mgreq.ini file in the engine's working directory, Magic automatically creates this server entry in the servers list: 'DEFAULT SOAP SERVER', with the type as 'SOAP'. You do not need to manually create a SOAP server entry.

Modal Window – rules

Below are three basic rules of modal GUI display forms

Opening Another Window

When you call a task from a modal window task, where the task form's modal property is set to Yes, the window of the called task will be opened as Modal, regardless of the new task window's modal property value.

Close window=No

A modal window task cannot keep the window opened after the task is terminated (Close window=No). A modal task window ignores the close window property and always treats this property as set to Yes.

Child Window=Yes

A child window cannot be modal. Magic will open any child window as modeless even if it is set to be modal or its parent task is modal.

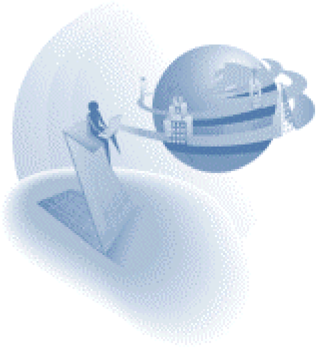
Temporary SQL Table Names

Temporary SQL table names are created for resident tables, direct SQL tasks (result tables), and for sort operations prefixed by `TEMP_`, as opposed to being prefixed by `T_.`

Task Generation - Skipping the Component's Main Program

When you execute the program of a Magic component, the component application's main program takes part in the Magic runtime task tree.

However, when you refer to a task's resources and information using functions like LEVEL, STAT or EOF the component's main program is not counted in the task generation. For example: if Task A calls Task B of a component, the function LEVEL(1) will refer to task A and not to the component main program.



Browser Client - Invoking Help

The Magic Help event is raised implicitly when you activate Help action on the browser, for example pressing F1 or CTRL+F1. When the Help event is raised a new Help window opens displaying the URL of the defined Help screen. The browser's help window will not be opened. If you request Help, and no help screen is assigned to the parked HTML control or is defined for the browser task form, the browser internal help window opens.

General characteristics of the Help window:

1. The Help window is a browser window designated for Help pages.
2. The Title of the Help window is always 'Help Page'.
3. The Help page window is opened in its last position.
4. When you click back on the task, the Help window remains open.
5. All help pages are opened in the same window. This means that the second help page opened replaces the content of the current Help page window and will not open a new window.

You can download a sample application that demonstrates this new feature from Browser_Client_Help_Activation.zip.

Browser Client - Integrating with External Modules

Magic can already call any external script using the CALLJS function.

Magic eDeveloper now also allows an external script to call Magic by raising a Magic event and to pass arguments to the event.

MGExternalEvent Function

A new function provided by the Browser Client JS module lets you invoke the Magic External event from an external script module that is executed on a Magic browser client page. Once this function is executed the External event is added to the Magic event queue. When Magic polls for events it handles the External event like any other Magic event. You can also use this function to pass arguments to the handler in Magic that handles the External event.

For more information on this function, refer to the eDeveloper Reference Guide.

You can download a sample application that demonstrates this new feature from Browser_Client_External_Integration.zip.

Browser Client - Mouse Wheel Support in Tables

The table control of a browser task will scroll through the records when the mouse wheel is used. This is only supported by Internet Explorer 6.

Browser Client - Calling Batch Tasks

The browser task supports calling batch tasks that send output to the requester IO. Calling a batch task that produces an HTML form or a merge form to a requester IO will result in displaying this output in a browser window as defined in the destination property of the Call program, Task or Exp operation.

Any destination frame defined in subsequent call operations to other batch tasks will be disregarded. Only the destination frame defined in the call operation to the batch task in the browser task will be used.

Batch Within the Current Context

The batch task that is invoked by a Call program, Task or Exp operation is run within the active context of the calling task. This means that all the context resources, such as variables, task tree, global parameters, memory tables, and transaction, are available to the called batch task. Any modification made to the context resources will be kept in the context when the batch task is completed.

Modeless Window

The window of the batch task's result output cannot be modal. It is always opened as a modeless window.

Result Window Out of Context

The result page opened when the batch task is completed will be out of the originating context.

Output from Nested Batch Tasks.

The output to the requester can be produced by the immediate batch task that is called from the browser task, and also by a batch task that is a descendant of the first called batch task. The output will be returned to the client only when the immediate batch task is completed.

No Output, No Window

If you run a batch program that produces no output to the requester, a page will not open.

You can download a sample application that demonstrates this new feature from Browser_Client_Calling_Batch.zip.

Browser Client - Flow Monitor and Context Change Notation

The Flow Monitor registers the change of a context and uses alternating background colors to distinguish between the contexts. See the eDeveloper Reference Guide for more information on the Flow Monitor.

CTRLGOTO - New Function

The CTRLGOTO function lets you park on a defined control as part of your application logic. See the eDeveloper Reference Guide for more details on this function.

You can download a sample application that demonstrates this new feature from CTRLGOTO.zip.

Property Sheet Automatic Handling

This new environment setting lets you decide if the property sheet should be opened and closed automatically depending on its relevance. See the eDeveloper Reference Guide for more details on this setting.

Application Server Priority

This new setting defines the priority of the Magic engine as an application server. See the eDeveloper Reference Guide for more details on this setting.

Mail Connection and Operation Timeout

Two new environment settings, Mail connection timeout and Mail operation timeout, set the timeouts for the various mail functions. See the eDeveloper Reference Guide for more details on this setting.

Select All Controls in Form Editor

The Select All form editor command selects all the controls of the form.

For a static control, if the form is in Edit mode, the Select All command will select the entire text of the control and not the form controls.

Select all Choice Control Layer Controls

Clicking a choice control, such as a Combo box, Radio button, or Tab & List box, while pressing the ALT key selects all the controls that belong to the current layer of the control.

Browser Task - Enabling Move/Repeat/Overwrite in HTML Controls

The Move entry, Repeat entry and Overwrite entry actions are enabled in the HTML control list of the Browser form.

Edit Control: Show Scroll Bars - New Property

The new Show Scroll bars property has been added to the Edit control of a GUI display form. This property determines whether to display a Horizontal scrollbar and a Vertical scrollbar according to the properties: Horizontal scroll and Vertical scroll.

Table Column - Unicode Support

Magic eDeveloper table columns can be defined as columns stored as Unicode. By setting the Character set to use (Char. Set) as Unicode, Magic automatically translates the Unicode data to ANSI data when the data is read, and translates the ANSI data to Unicode data when the data is written to the database.

Get Definition

If the SQL type is of Unicode, the Char. set property is set to Unicode when getting the definition of a table.

Creating a Table

If no SQL type is defined and you are creating a table with columns set to Unicode, the column will be set as NCHAR.

Note: The Unicode option is supported for Microsoft SQL Server® and Oracle®.

F11 and F12 Keys Support

Magic now also supports the F11 & F12 key combinations. The supported combinations are: F11, F12, CTRL+F11, CTRL+F12, ALT+F11, ALT+F12, SHIFT+F11, and SHIFT+F12

XML Direct Access Support

You can use Magic's new XML functions to easily access XML documents via a Magic application. These functions let you:

1. Receive the data of the various XML objects using the XMLGET function.
2. Check the existence of an XML object using the XMLEXIST function.
3. Find an XML object using the XMLFIND function.
4. Count the occurrences of a given XML object using the XMLCNT function.

For more information on the syntax of the XML functions and how to use them see the eDeveloper Reference Guide.

You can download a sample application that demonstrates this new feature from XML_Direct_Access.zip.

Form Repository: Child Window Property - Enhancement

The Child Window property has been modified so that it can be defined by an expression. You can only set an expression for this property from the property sheet and not from the child window column. This property expects a Boolean return value. The expression is computed when the form is opened before the task prefix.

Hyperlink Dialog - Enhancement

The Hyperlink dialog has been modified and now lets you define expressions for the following fields: System, Program, and Destination.

Modified Decimal Separator Support

When you set this Magic Specials flag as follows:

```
[MAGIC_SPECIALS]
SpecialModifiedDecimalSeparator = Y
```

Magic returns the decimal separator character defined in the Magic.ini file whenever the point key is pressed, either on the main keyboard or on the number pad, only when you are parked on a numeric field.

Note: The replaced decimal separator character is defined in the Magic environment and not as defined by the machine's regional settings.

Web Services Consumer - Call Web Service Assistant

The Call Web Service operation helps you query the available methods of a given web service (WSDL) to automatically produce the desired call operation. For more information on Web Services support see the eDeveloper Reference Guide.

Web Services Consumer – HTTPS support

The Call Web Service operation now supports web services provided over a secured HTML transfer protocol (HTTPS)

Windows Compiled HTML File Support

The Windows Help entry type now also supports the Windows Compiled HTML help format: .chm. To use a Compiled HTML file just enter the .chm file name and location in the Help properties.

Note: Only these HTML commands are supported: Context, Contents and Quit. The other HTML commands are regarded as Contents.

Directory Binding String - LDAP Support

The Directory Binding string, found in Settings/Environment/External, which used to work with directory services for automatic logon, now supports LDAP.

The Directory Binding string is used to bind into the directory service. During the binding process, Magic automatically concatenates the operating system username to the string and attempts to bind into the operating system.

The syntax for LDAP binding is as follows:

```
LDAP://HostName[:PortNumber][/DistinguishedName]
```

The syntax for Win32 binding is:

```
WinNT:[//DomainName/[ComputerName/[ObjectName[,className]]]]
```

eDeveloper 9.3 Service Pack 3

Important Notes

Enhanced Security File

The Magic security file has been enhanced to allow longer user names and passwords. The new structure of the security file provides a 30-character long User ID and a 20-character long Password.

If you use the new security file structure, you cannot use security files from previous versions with Version 9.3 SP3.

Converting an existing security file to the new structure

Magic eDeveloper 9.3 SP3 provides a command line utility that lets you convert an existing security file to the new structure. The name of this utility file is Usrupd.exe.

The usage syntax is:

```
usrupd.exe <input filename> <output filename>
```

where: `input filename` is the original security file name, and `output filename` is the new security file name

Team Development

If you develop your Magic application in Team Development mode, note that the Team Development mechanism does not support User IDs longer than 10 characters.

Interactive Web Application – Known Problems

Incorrect Status of Progress Bar

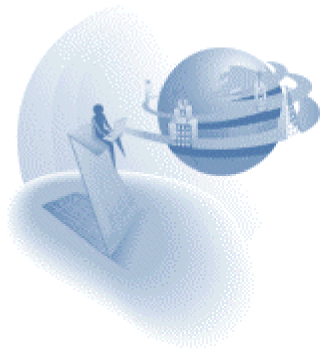
The Internet Explorer progress bar may show an incorrect status when pages are opened. This is a known problem of Internet Explorer that has been acknowledged by Microsoft.

Void Hyperlinks Over Slow Connections

Hyperlinks that require no actual URL may usually be set to open a void URL using the “javascript:void(0)” script. When such a hyperlink reference is set for the hyperlink directly on the HTML page, the client may get stuck if these hyperlinks are clicked while the browser client page is loading over slow connections. To avoid this, it is best to define the “javascript:void(0)” hyperlink in the Hyperlink property of a Hypertext control in the Magic toolkit instead of defining the hyperlink directly on the HTML page using the HREF attribute.

NSAPI Requester – Cease of Support

Due to the fact that there is no reported use of the NSAPI requester, starting from Magic 9.3 Service Pack 3 the NSAPI requester will no longer be provided and starting from Magic 9.3 Service Pack 4 the NSAPI requester will no longer be supported.



Behavior Modifications

Removing the System Logon FULL Option

The Full option of the System Logon environment setting has been removed.

Directory Binding String

The Directory Binding String is required to direct the Magic engine to the relevant directory service when the System Logon environment setting is set to Directory.

This string is no longer defined in the Magic environment, and it must be set by a secret name of the security file. The Directory Binding String should be defined as the translation of the reserved secret name `Directory_Binding`.

Important note: When you set the System Logon environment to Directory, no manipulation of the Magic security file is allowed.

Browser Client – Nested Deferred Transactions

A browser client task opened in a modeless state, either by setting its Modal property to No or by opening it in a frame, cannot have its transaction mode set to Nested Deferred. Only the options Deferred or None can be set for a modeless browser client task.

Browser Client - Task Suffix Execution

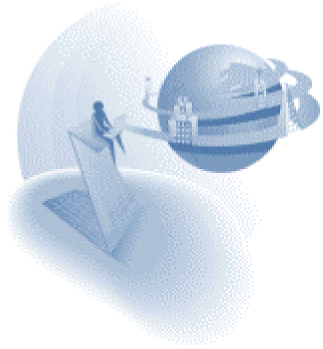
The task suffix is now executed on the client side instead of on the server side, as in previous versions.

Browser Client – Default Chunk Size

When no chunk size expression is defined in a browser task, Magic sets a default chunk size value according to the following rules:

If no Main table is defined for the task, the default chunk size value is set to 1.

If a Main table is defined for the task, the default chunk size value is set to 30.



New Features

Browser Client – Handling IFRAME in Runtime

The Magic browser client modules fully support the execution of browser tasks within an IFRAME control.

Browser Client - Indication of Server-Side Activity

The running browser task indicates server-side activities by changing the cursor to an hourglass shape. The cursor shape is changed back to the default when the server-side activity is completed.

Browser Client - Support View Refresh and Relocate in Subforms

Up until 9.3 SP2, when one of the variable arguments was modified in a recomputed subform, the subform would only refresh its view according to the new range criteria. Starting from 9.3 SP3, the subform also relocates a record according to the new locate criteria when it is recomputed.

Transaction Mode – None

A new option of None has been added to the Transaction mode task property. See the eDeveloper Reference Guide for more details on this option.

Browser Client – Additional Browser Client Module

The browser client modules have been enhanced by an additional COM object module. The COM object module, named Mgbccom9.cab, is provided in the Scripts directory of your installation. This additional COM object module greatly enhances the performance of browser client tasks.

The COM object is automatically added to the browser when it opens the browser client application for the first time. The browser issues a confirmation dialog that requires the end-user to confirm using the COM object.

eDeveloper 9.3 Service Pack 4

General Notes

Intersystems™ Caché™ DBMS is not supported

Magic eDeveloper no longer supports the Caché DBMS.

The Caché gateway and the installation are no longer provided.

A Magic application can still work with Caché databases through ODBC.

APPOWER – Module Removed

The APPOWER utility that accompanied the Magic eDeveloper toolkit is no longer supported. The installation of Magic eDeveloper Version 9.3 SP4 will remove previous installations of the APPOWER utility

New HTML Help system

The Magic eDeveloper's internal help system is now in CHM format. The help system's updated interface provides enhanced search capabilities.

The Magic installation will automatically update your Magic.ini file to the location of the updated help file. If the default Magic.ini file is not used, set the Help file environment setting, located under `Settings\Environment\External Files`, to the Mghelpw.chm file.

Behavior Modifications

Window Display of Program Opened Through Parent Task Handlers

In Previous versions of Magic eDeveloper the child window of programs that were called from a parent task handler were not displayed and handled correctly. The child window of a program called form a parent task handler used to be handled as a direct descendant to the parent task window instead of being handled as a direct descendant to the task that raised the event. This behavior has been corrected in Magic eDeveloper 9.3 SP4 so programs that are called from a parent task handler are handled as direct descendants to the task that raised the event.

The corrected behavior of Magic eDeveloper 9.3 SP4 may affect existing application that relied on the previous wrong behavior. The effect of this change may be child window programs that appear either as being displayed more times than expected or as being displayed in a different location than expected.

You may download a sample application that demonstrates the corrected behavior and its effect using this link: 020_ChildWindows.exe

Floating Windows Startup Position

In previous Magic eDeveloper versions, floating windows were always opened as centered regardless of the Top and Left properties of the form.

Floating windows could only open as defined by the form's navigational properties if the following flag was entered to the Magic.ini file under the [MAGIC_SPECIALS] section:

```
[MAGIC_SPECIALS]
SpecialCenteredFloatingWindow=N
```

Along with the Startup Position, a new GUI Display form property, the `SpecialCenteredFloatingWindow` default setting has been changed to No.

If your application depends on Magic to automatically center all floating windows, you can have the `SpecialCenteredFloatingWindow=Yes`.

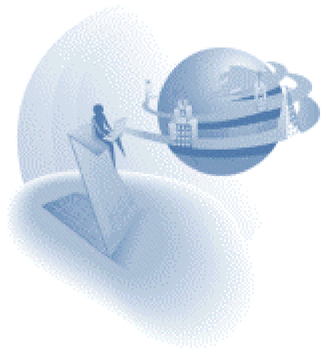
If you already have the `SpecialCenteredFloatingWindow=No`, your application will not be affected. You can remove this flag from the Magic.ini file.

Shared Resident Table Support for the Application Server

In previous Magic eDeveloper versions, the background application server engine loaded the resident tables of the same application for every new engine context.

A table is loaded as a resident table is in the read-only access mode, where the same instance of the resident table can be shared by all contexts handled in the same application server.

In Magic eDeveloper 9.3 SP4, each resident table is loaded only once and is shared by all engine contexts that are handled in the same application server process. This modification greatly improves the performance of applications that use resident tables and reduces the use of memory resources.



Active-X Control – Loading and Handling

In previous Magic eDeveloper versions, OLE controls that were set to display Active-X classes were handled by Magic as an OLE control and not as an Active-X control. This caused the following behavior:

1. The Active-X control was not automatically loaded. It could only be loaded through the Insert Object operation or by updating the control by using a pre- defined Blob data item containing the Active-X persistent data.
2. The Magic runtime engine occasionally crashed due to handling an Active-X class as an OLE control.
3. The end user could not activate the interface of an Active-X control.

These problems have been fixed in Magic eDeveloper Version 9.3 SP4.

Important Note: Active-X controls that were initialized by a pre-defined Blob data item containing Active-X persistent data cannot be used in SP4. You should remove the initialization sequence by disabling the update operation or by simply removing the file by which the blob variable is updated.

Default Settings for New Batch Tasks

Most batch tasks usually have the Allow Options property and the Open Task Window property manually modified from Yes to No.

Now the default value of this property has changed from Yes to No.

The Link Left Outer Join Operation Supports the ISAM DBMS

The Link Left Outer Join operation is relevant only to RDBMS and is not specifically relevant to the ISAM DBMS, and therefore Magic did not define the Link Left Outer Join operation for tasks that handle ISAM tables.

Magic eDeveloper now lets you define the Link Left Outer Join operation for tasks that handle ISAM tables as a Link Query operation.

This updated feature makes Magic eDeveloper more flexible when working with different DBMS systems, and enables you to define the Link operation of a task to work as Left Outer Join for RDBMS tables and as Link Query for ISAM tables.

Form Display of a Closed Task

When a task in the Create mode is closed while keeping its window opened can be instructed to either compute the initial and default values of the variables and the expressions of the controls or not.

A new special flag provides the two modes of handling a form of a task as described above.

```
[MAGIC_SPECIALS]
SpecialPaintFormInCreate= Y/N
```

When this flag is set to Yes, all the variables’ initial and default values are computed and all the controls’ properties based on expressions are computed as well.

When this flag is set to No, all the variables’ initial and default values are not computed and all the controls’ properties based on expressions are not computed as well.

The default value of this special flag is No.

New Features

XML Component Generator

Magic eDeveloper lets you handle any XML file for which an XML schema exists. Open the XML Component Generator tool by selecting the XML Generator option, from the Options\Components menu, to create a component application that can access and handle XML information.

For more information about the XML Component Generator, please refer to the XML Component Generator chapter in the Reference Guide.

Note: The XML Component Generator of 9.3 SP4 is not yet supported for a right to left oriented toolkit.

Variable Selection optimization

A variable list, opened in a task, will park on the first variable of the current task if no other variable is selected.

Property Sheets Support Marked Rows

Magic eDeveloper lets you handle the properties of objects listed in a table and supported by the property sheet. Such objects are Models, Columns, Local Variables and HTML Controls.

You can set the same property value to several columns by multi-marking the relevant column rows for the value set in the property sheet.

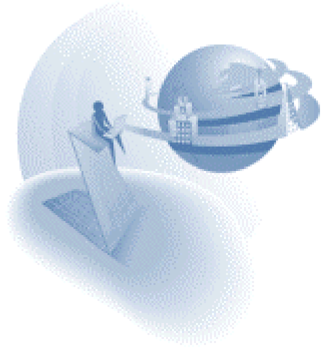
Direction Option in the Flow Monitor Find Dialog Box

The flow monitor Find dialog box lets you set the direction by which the flow monitor searches for a string.

Control Change – New Event

The Control level handler for online programs supports the Change event. The Change event is triggered when the end user modifies the control value, and lets you better handle each task field and respond to the end user’s input. The Control Change event is not supported in browser tasks.

You may download a sample application that demonstrates this new feature using this link: 017_Control_Change.exe.



ASTR – New Function

The ASTR function lets you apply an alpha picture to a string value.

For more information about the ASTR function, please refer to the Expressions chapter in the eDeveloper Reference Guide.

You may download a sample application that demonstrates this new feature using this link: 015_ASTR.exe

HTTPS Support in HTTPPOST & HTTPGET functions

The HTTPPOST and HTTPGET functions now support transmissions over a secured HTTP.

For more information about the HTTPPOST and the HTTPGET functions please refer to the Expressions chapter in the eDeveloper Reference Guide.

Authentication Support for the HTTP Proxy Environment Setting

The HTTP Proxy environment setting, under Settings/Environment/External Files, is required when using the HTTPPOST function, the HTTPGET function and Web Services.

If your proxy server requires a user name and password for authentication, you can enter the user name and password in the proxy setting string in the format below:

```
[username]:[password]@[Proxy Server IP\Host name]:[port number]
```

For example: myuser:mypass@10.1.1.1:8080

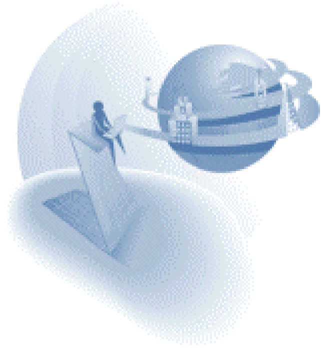
For more information about the HTTP Proxy setting, please refer to the Settings chapter in the eDeveloper Reference Guide.

RQHTTPHEADER - New Function

The RQHTTPHEADER function lets you modify header tokens of any HTTP result. It is primarily used to modify the content type of the result.

For more information about the RQHTTPHEADER function, please refer to the Expressions chapter in the eDeveloper Reference Guide.

You may download a sample application that demonstrates this new feature using this link: 012_RQHTTPHEADER.exe



HTTPLASTHEADER - New Function

The HTTPLASTHEADER function lets you retrieve the header information of the last HTTP result by using either the HTTPPOST function or the HTTPGET function.

For more information about the HTTPLASTHEADER function, please refer to the Expressions chapter in the eDeveloper Reference Guide.

VARDBNAME - New Function

The VARDBNAME function lets you retrieve the physical name of a real variable, along with the physical name and the owner name of the table.

For more information about the VARDBNAME function, please refer to the Expressions chapter in the Reference Guide.

GUI Form - Placement on Dynamic Dimensions

Previous Magic eDeveloper versions did not let you define a placement factor and an expression to determine the dimension of a GUI Display control.

In Magic eDeveloper Version 9.3 SP4, this restriction has been removed. When a control’s dimension is defined by an expression and a placement factor, Magic automatically handles the expression modification by adding it to the size determined by the placement factor for the form in runtime.

Note: The automatic computation of both the dynamic expression and the placement factor is available only for controls that are direct descendants of the form.

TRANSLATE - New Function

The TRANSLATE function lets you translate a string that contains logical names. The TRANSLATE function also supports the translation of logical names that are nested.

The TRANSLATE function resolves the issue of having to use the recursive INIGET operation for resolving logical names.

For more information about the TRANSLATE function, please refer to the Expressions chapter in the Reference Guide.

MAILBOXSET - New Function

The MAILBOXSET function lets you specify the mailbox for retrieving messages.

For more information about the MAILBOXSET function, please refer to the Expressions chapter in the Reference Guide.

MAILLASTRC - New Function

The MAILLASTRC function lets you retrieve the return code of the last mail function.

For more information about the MAILLASTRC function, please refer to the Expressions chapter in the Reference Guide.

Optimized Image Loading

The Magic eDeveloper runtime client has been optimized to cache images to memory, so recurring references of an already loaded image will have the image loaded from the memory and not from the disk.

You can control the maximal memory size allocated for caching images from the Image Cache Size environment setting, located under Settings/Environment/Preferences.

You can set the Check Image Change Time environment setting to Yes to have Magic check the time stamp of each image before it is loaded so updated image files will take effect.

The Check Image Change Time environment setting is located under Settings/Environment/Preferences.

For more information about the Image Cache Size and the Check Image Change Time environment settings, please refer to the Settings chapter in the Reference Guide.

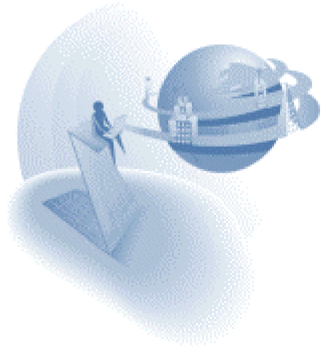
Multi-Marking Enhancements

Magic eDeveloper's Multi-Marking mechanism has been enhanced by the new Multi-Marking events below:

- **Mark All** – Marks all the records of a task.
- **Mark to Bottom** – Marks all the records from the current record to the last record of the task.
- **Mark to Top** – Marks all the records from the current record to the first record of the task.
- **Invert Marking** – Marks all the records that are not marked and unmarks the records that are marked.
- **Unmark All** – Unmarks all the records of the task.

Note that raising the events of Mark All, Mark to Bottom, Mark to Top, and Invert Marking may require loading the entire task dataview. Raising these events in tasks with a very large dataview may take a considerable amount of time.

You may download a sample application that demonstrates this new feature using this link: 016_MultiMarking.exe



Directory Service Support – Enhanced Logon Functionality

Previous versions of Magic eDeveloper have provided the functionality for logging into Magic by using the operating system's logon state and by retrieving the user's assigned group from the Directory Service to determine the user's rights.

This functionality has been further enhanced in Magic eDeveloper 9.3 SP4 and supports all Logon functionality that can be performed for a defined Domain to further retrieve the roles of a user through a defined Directory Service.

You can set Magic eDeveloper to use a Directory Service from the System Logon environment setting, located under Settings/Environment/System.

When in the Directory mode, any logon procedure (such as the Logon dialog, the Logon function or logging on through a request) will be compared to a defined Domain, and the roles will be retrieved from a defined Directory Service.

For more information about the Directory mode and how to set the Domain Name and the Directory Service, please refer to the Settings chapter in the Reference Guide.

Startup Position – New GUI Display Control Property

The Startup Position property lets you set the task window to open in various pre-defined and automatically computed locations.

The Startup Position property is located in the Navigation section of the GUI Display Form control property sheet.

For more information about the Startup Position property, please refer to the Display Form chapter in the Reference Guide.

Fit to MDI - New GUI Display Control Property

The Fit to MDI property lets you expand a task window to the entire client area of the Magic window.

The Fit to MDI property is located in the Navigation section of the GUI Display Form control property sheet.

For more information about the Fit to MDI property, please refer to the Display Form chapter in the Reference Guide.

You may download a sample application that demonstrates this new feature using this link: 013_Fit_to_MDI.exe

I/O Device - Media of Variable

The Variable option is used to send and receive a textual variable for output and input forms. This option provides for the storage and exchange of XML files.

For more information about the Variable media type, please refer to the Programs chapter in the Reference Guide.

Quit - New Event

The Quit event combines both the Cancel and Exit event.

Note that for this eDeveloper release, the Quit event is not supported by browser client tasks.

For more information about the Quit event, please refer to the Magic Help file.

Magic Requester – Additional Settings

Two settings have been added to the MGREQ.INI file that affect the execution of the requester:

RetryMainTime=XX (default: 5)

The RetryMainTime determines the time interval for a requester, connected to an alternating Request Broker, to try reconnecting to the main Request Broker.

If the requester is already connected to the main Request Broker, the RetryMainTime setting does not apply.

The time interval is specified in minutes.

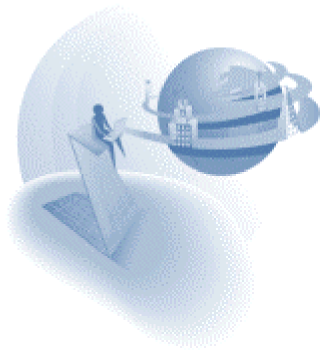
For example, when RetryMainTime=15, the requester tries to reconnect to the main Request Broker every 15 minutes. When the main Request Broker becomes available, the requester will connect to it and will disconnect from the alternate Request Broker. If the main Request Broker becomes unavailable, the requester returns to the alternate Request Broker and tries to reconnect to the Main Request Broker every 15 minutes.

When RetryMainTime=0, the requester will not try to connect to the main Request Broker, but will remain connected to the alternate Request Broker.

KeepAlive=Y\N (default: N)

When KeepAlive=Yes, the Magic Generic Messaging Layer module (mgrqgnrc93.dll) lets the system administrator use operating system level settings to control the keep alive intervals for each open connection.

Only a system administrator should determine the settings for the operating system intervals.



Default Storage – New property

Magic eDeveloper 9.3 SP4 enables you to easily set table columns to be automatically mapped to the expected storage of the underlying DBMS according to the column's attribute and the DBMS gateway in use. This ability adds to the transparency of DBMS portability.

A Default Storage new property is added to table columns. This property is located under the Storage section. The Default Storage property may be set by Yes or No values. When the Default Storage property is set to Yes the column will be mapped according to its default mapping that is determined by the DBMS gateway in use. When the Default Storage property is set to No the column will be mapped according to the column's Storage property. The Default Storage property is also available for virtual variables used in Direct SQL tasks.

The default value of the Default Storage property for newly created columns is No.

The default value of the Default Storage property for Direct SQL task's virtual variables is Yes.

On importing application export files of previous versions, the Default Storage property will be set to Yes if the values of the Stored As, Size, and Definition properties are the same as the default values defined by the underlying DBMS gateway that is set for the table, otherwise the Default Storage property will be set to No.

IFRAME Browser Control Support

Magic eDeveloper lets you use the IFRAME control to handle IFRAME elements on your HTML page.

For more information about the IFRAME control, please refer to the Display Form chapter in the Reference Guide.

Opaque Browser Control Support

Magic eDeveloper lets you use the Opaque control to handle other HTML elements that are not explicitly handled by Magic eDeveloper as a control in the Magic toolkit.

For more information about the Opaque control, please refer to the Display Form in the Reference Guide.

CALLURL and CALLPROGURL – New Functions

The CALLURL and CALLPROGURL functions let you activate either a simple URL or a URL that calls another Magic program as a request, which is external to the current browser client context.

For more information about these functions, please refer to the Expressions chapter in the Reference Guide.

Optimized Server-Side and Client-Side Handling

The Magic operations and functions are categorized into three categories:

- **Client** – The operations and functions that must be executed on the client. The CALLJS function is an example of a client-side function.
- **Server** – The operations and functions that must be executed on the server. The INIGET function is an example of a server-side function.
- **Client and Server** – The operations and functions that can be executed on either the client or server.

The browser client module has been enhanced to optimize the handling of different operation categories by executing the operations, with their functions, for either the client or server, as determined by the next operation.

For example, when a browser client turns to the server to execute a server-side operation, the browser task will continue to execute the handler operations until it encounters a client-side operation.

This mode of execution is performed per handler.

You can view the category of an operation in the toolkit when browsing the task handlers and Operation list. The operation category is displayed on top of the first column in the Execution table. You can turn off the category display by not selecting the Show Handling Info option in the Options menu.

You can also use the Show Handling Info option to display the function category of each function in the Function list.

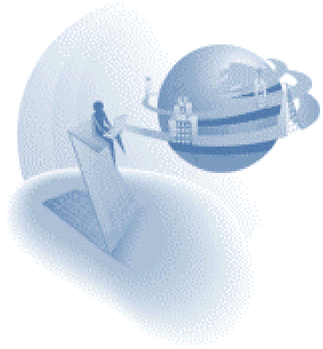
Note that you can save repeated server-side interaction by placing server-side operations one after the other.

Localization of the Browser Client Module

In previous versions of Magic eDeveloper, all internal runtime messages were in English.

Starting Magic eDeveloper 9.3 SP4, all runtime messages for the active browser client application are displayed in the language specified for the Magic application server by the active mgconstw file.

The MGBCL930_04XXX.cab has been added to the Browser Client Module file to store locally handled messages. When installing Magic eDeveloper, each selected language will install the required language support file.



Additional Preset Tool Images

In previous versions of Magic eDeveloper, two preset tool images where supported, the MG_VCR image and the MG_EDIT image. By placing these images on the HTML form, the browser client could handle the images automatically and trigger the relevant internal events.

Magic eDeveloper Version 9.3 SP4 supports two additional preset tool images, MG_CANCEL and MG_EXIT.

By placing the images that are named MG_CANCEL and MG_EXIT on the HTML form, the browser client will automatically trigger the Cancel event when the user clicks on the MG_CANCEL image and the Exit event when the user clicks on the MG_EXIT image.

The Magic installation provides the mgcancel.jpg and the mgexit.jpg sample images in the Scripts directory image.

eDeveloper 9.3 Service Pack 5

Behavior Modifications New Features

Requester – Handling Numeric Arguments

Numeric arguments passed to a Magic program using a requester were handled in Magic eDeveloper 9.3 SP4 as pure numeric values.

Magic eDeveloper 9.3 SP5 has reverted the way of handling numeric arguments to have them passed as strings.

The Magic Application Server automatically assigns the passed values to String, Numeric, Logical Date, or Time values.

Handling the Numpad Plus Key

Magic applications that are required to handle the Numpad Plus key may select it as an event in the task handler or in the Magic keyboard mapping file.

You should note that even though the Numpad plus key may be mapped to an internal event or used as a trigger, the Magic task still receives the plus character as keyboard input.

You can instruct the Magic engine to ignore the plus character by setting a Magic special flag as follows:

```
[MAGIC_SPECIALS]
SpecialNumpadPlusGetCharacter=N
```

Setting this special flag to No affects the Numpad plus key at all times, even if the key is not handled or mapped to any internal event.

eDeveloper 9.3 Service Pack 5a

Fixed Problems

Abort on Object Check In (QCR 755159)

The eDeveloper engine crashed on aborting a Check In action on a locked object when the application table is stored as an MSSQL table.

SQL Assistor Crash (QCR 992885)

The eDeveloper engine crashed on scrolling inside the SQL assistor when the 13th table entry was of a different database than the preceding table entries.

eDeveloper 9.3 Service Pack 5b

Fixed Problems

Improper Record Fetch in Link L.O Join I (QCR 423826)

Records where improperly retrieved if the task had Link Left Outer Join operations and each has its Evaluate Link Condition property set differently.

Improper Record Fetch in Link L.O Join II (QCR 427470)

Records where improperly retrieved if the task had several Link Left Outer Join operations.

Group on Updated Virtual Field (QCR 751558)

Magic crashed on executing a batch task set with a group level of a virtual field and in that group level the same virtual was updated.

Crash on a Group level and Link L.O. Join (QCR 760304)

Magic crashed on executing a batch task set with a group level and defined with Link Left Outer Join operations.

Failure on Connecting to Broker (QCR 762484)

A Magic Server engine failed to connect to the Magic Broker if the startup application had a force-write INIPUT function in its Task Prefix handler.

Program with Link Join and Record Lock Aborts (QCR 764333)

A program used to abort with an error message of 'Record has been updated' if the program was defined with Link Join operation and with a “On Record Lock” physical transaction.

Thumb of Table’s Scrollbar Not Properly Positioned (QCR 787928)

In a GUI Table control, no reasonable space was kept between the thumb of the scroll bar and its bottom end as records were scrolled through.

Dynamically Set Forms (QCR 984011)

Batch task forms where not positioned properly according to their dynamically set navigation properties.

eDeveloper 9.3 Service Pack 5c

Fixed Problems and New Feature

Failing to Connect to the Broker (QCR 751047)

The Magic engine could not connect to the Magic broker if the Magic engine was started by the Magic Broker and the TCP/IP settings included /LocalHost option.

Hebrew Translation of the Quit an Exit (QCR 974685)

The Quit action and Exit events were translated to the same name in Hebrew - Exit.

Negative Values passed in Oracle DSQL (QCR 765901)

Negative signs were lost when passing arguments as expressions in an Oracle DSQL task.

Mail Message Character set

A new entry in the Magic.INI enables you to set the character set required for sending mail messages using the MailSend function.

Set the required character set name in the following in the Magic.INI as follows:

```
[MAGIC_ENV]
MailCharset =
```

If no character set is defined Magic the default character set is US-ASCII.

eDeveloper 9.3 Service Pack 5d

Fixed Problems

Java Exception in Browser client application

Data controls were in some cases improperly handled causing the flow of the client to fail.

Toolbar Display on Windows XP (QCR 292592)

The Toolbar of the Magic client was not properly displayed on Windows XP when opening large applications.

Improper Records Fetch of Links L.O. Join (QCR 452470)

Data view constructed of Link Outer Join operations did not produce all expected records if the Evaluate link condition settings of the link operations was not the same

eDeveloper 9.3 Service Pack 5e

Fixed Problems

"FM_TRANS Error (56)" error (QCR 286105)

The eDeveloper Enterprise Server occasionally produce an error message labeled: "FM_TRANS Error (56)"

Memory Leak on Using Resident Tasks (QCR 777310)

Executing an eDeveloper application using an Oracle database caused a memory leak when the application's tasks were set to be resident.

Empty Toolbar Not Removed (QCR 989053)

A blank toolbar would appear when no toolbar options are defined instead of being removed.

eDeveloper 9.3 Service Pack 5g-h

Fixed Problems and New Feature

Request Argument Improperly Retrieved (QCR 441444)

Arguments of a remote request were not retrieved correctly if the eDeveloper server engine was engaged by an already running online task.

Crash on Passing a Blob via Call Web Service (QCR 755681)

Magic crashed on executing a Call Web Service operation and a blob was sent as a parameter using the FILE2BLB function with a non-existing file.

Improper Import of 8.3 Export file (QCR 917410)

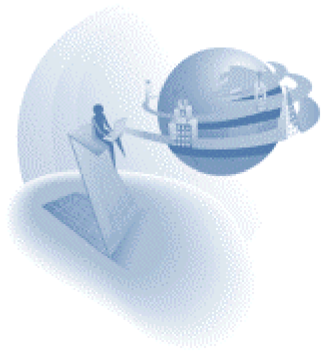
8.30 export file was not imported correctly in 9.30SP5d. There was a syntax checker error for combo box label ranges.

Threads Count

A new thread counting mechanism has been introduced between the broker and the enterprise servers, in order to eliminate a state where certain engines were in a ‘busy request’ state although they weren’t busy.

In order to activate this mechanism, you need to set the following new entry in the MGRB.INI file:

```
CountersEqualizationDuringStatus  = Y
```



eDeveloper 9.4

Behavior Modifications

New Licenses

eDeveloper Version 9.4 requires new licenses.

For more information on the new license and how to acquire it, please refer to the Guide to Installation and License Management

Mixed Case Function Names

The function names have been changed from upper-case-only strings to mixed-case strings.

Functions of export files from previous versions are imported properly.

Optimizing Environment Settings Management in Enterprise Server Engines

When a context is created on the Enterprise Server, it loads a copy of the Magic.ini file. Each context keeps its own copy of the Magic.ini file, which means that changes made to the INI file using an INIPUT operation that forces the entry to be written in the INI file will only be loaded by new contexts.

Optimizing User File Management in Enterprise Server Engines

In previous versions whenever a user-related functionality was required, the engine would read the users file directly from the disk or update it.

Starting with eDeveloper 9.4, the users file is loaded to the memory of the engine process, eliminating the IO interaction.

Any modification of the users list, such as evaluating the UserAdd function, results in updating the users file with the updated users list.

The process re-loads the users file every 60 minutes so that it can be updated with any modifications to the users list that might have been introduced by a different engine.

Modifying Location and Size of eDeveloper Window

In Version 8, you could affect the location and dimensions of the eDeveloper window using INI settings: MDIleft, MDItop, MDIright, MDIbottom

Starting from Version 9.00, the location and dimensions of the windows have been kept in the registry, which did not allow these settings to be automatically modified.

In eDeveloper 9.4 you can override these settings by making additions to the Magic.ini file in a similar way to the way it was done in V8. Any modification introduced to the location and dimensions will still be updated and kept in the registry.

Component Builders Modifications

The Component builder utility of eDeveloper previous versions has been divided into separated builders, where each builder is dedicated to a specific interface (eDeveloper interface, WSDL, EJB and the new COM interface).

The data that was already entered in the previous builder should be separated and converted to enable the new builders to properly handle this data.

If you wish to keep the component builder data of previous versions you should download a conversion utility.

XML Component Generator Modifications

The structure of the information stored by the XML Component Generator has been modified.

The data that was already entered in the XML Component Generator of a previous build should be converted to allow the new XML Component Generator to properly handle this data.

If you wish to keep the XML Component Generator data of previous versions you should download a conversion utility.

New Features

New Application Option

You can easily create new applications using a New Application dialog available from the New option on the File menu.

For more information on the New Application option, refer to the eDeveloper Reference Guide.

COM Object Support

eDeveloper provides you with full functionality to handle OLE objects and Active-X controls.

For more information on how to handle COM objects, refer to the eDeveloper Reference Guide.

COM Interface

Any eDeveloper application programs can be executed as COM object methods using a COM interface.

eDeveloper provides a COM interface builder that lets you easily create a COM interface for your applications.

For more information on the COM interface builder, refer to the eDeveloper Reference Guide.

Drag and Drop Support

You can easily provide Drag and Drop functionality in your applications using the new Drag and Drop support. With eDeveloper 9.4 you can Drag and Drop between an eDeveloper application and other applications, and within a single eDeveloper application.

For more information on the Drag and Drop functionality, refer to the eDeveloper Reference Guide.

Tree Control Support

You can easily display hierarchical data in a Tree format using the new integrated data-bound Tree control for GUI forms.

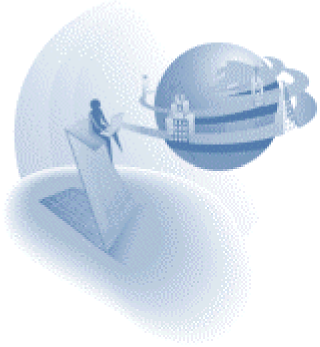
For more information on the Tree control, refer to the eDeveloper Reference Guide.

MSMQ Messaging Capabilities

You can use the eDeveloper Messaging component to take advantage of messaging capabilities via a Microsoft Messaging Queue server.

Note that currently Peek or View Access to MSMQ is not supported.

For more information on the Messaging features, refer to the Magic's Messaging Connectivity Components document.



JMS Messaging capabilities

You can use the eDeveloper Messaging component to take advantage of messaging capabilities via any messaging server that supports JMS.

For more information on the Messaging features, refer to the Magic's Messaging Connectivity Components document.

MQ Series Messaging capabilities

You can use the eDeveloper Messaging component to take advantage of the messaging capabilities via the MQ Series messaging server.

For more information on the Messaging features, refer to Magic's Messaging Connectivity Components document.

SNMP Monitoring Support

The enterprise server inherently supports SNMP monitors sending SNMP trap messages. You can also have your application send application trap messages using SNMP related functions.

For more information about SNMP in Magic, refer to the eDeveloper Reference Guide.

Java Objects Support

You can handle and interact with any Java object using a collection of eDeveloper functions.

eDeveloper provides a Java Access generator that easily lets you generate an application that handles a selected Java object.

For more information about Java related functions and the Java Access generator, refer to the eDeveloper Reference Guide.

Call Public

A new Call operation mode lets you define a dynamic call to a program by its public name. The called program can be a program in the current application or in another application.

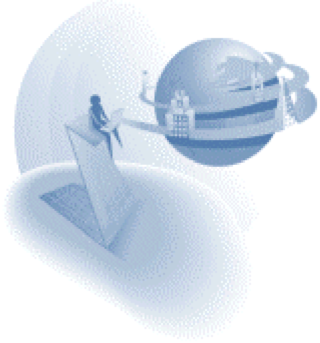
For more information about the Call Public operation, refer to the eDeveloper Reference Guide.

Public Events

You can set user-defined events in the Main Program to be exposed to other applications that are loaded as components.

Exposed user-defined events can be raised from a loaded component using the Public Event new type of the Raise event operation.

For more information about Public Events, refer to the eDeveloper Reference Guide.



Block Loop

There is a new Block operation mode that lets you create cyclic sections of operations.

For more information on the Block loop operation, refer to the eDeveloper Reference Guide.

Print Data

This new feature provides the end-user the ability to output the data of the current view in various formats.

For more information on the Print Data capability, refer to the eDeveloper Reference Guide.

LDAP Integration

eDeveloper provides enhanced authorization capabilities using LDAP services.

For more information about LDAP connectivity and LDAP-related functions, refer to the eDeveloper Reference Guide.

Encryption Functionality

eDeveloper provides new functions to cipher and decipher data.

The encryption-related functions are Cipher and Decipher.

For more information about these functions, refer to the eDeveloper Reference Guide.

Browser Client - Supporting Parallel Transactions

Starting with this version, browser-client applications can handle several transactions at the same time.

You can use a browser client application to open a nested deferred transaction that will be independent of its ancestor transactions.

Browser Client - Caching the Subform View

Version 9.4 provides further optimization of the browser-client technology by enabling you to set the view of the subforms to be cached locally on the client side.

For more information refer to the documentation on the Is Cached property of the subform control in the eDeveloper Reference Guide or the eDeveloper Help.

Memory Table Persistence

Using two new functions, MTblGet and MtblSet, you can keep the content of memory tables persistent so that the memory-table content can be stored and transferred as needed.

For more information about these functions, refer to the eDeveloper Reference Guide.

Vector Support

A new set of functions lets you handle vector fields that allow you to pass and receive vector values between applications and third party applications.

The vector-related functions are VecSet, VecGet, VecCellAttr, and VecSize.

For more information about these functions, refer to the eDeveloper Reference Guide.

Buffer Management

A new set of functions lets you handle buffer content that can be communicated with legacy applications. The buffer management functions are:

BufGetAlpha, BufGetBit, BufGetBlob, BufGetDate, BufGetLog, BufGetNum, BufGetTime, BufGetVariant, BufGetVector, BufSetAlpha, BufSetBit, BufSetBlob, BufSetDate, BufSetLog, BufSetNum, BufSetTime, BufSetVariant, BufSetVector.

For more information on these functions refer to the eDeveloper Reference Guide.

Remote Flow Monitor

The remote flow monitor has been enhanced with more options to control the varying context of a background server.

For more information on the remote flow monitor, refer to the eDeveloper Reference Guide.

Enhanced Checker

eDeveloper offers an enhanced syntax checker that displays the result messages in a separate floating and dockable window, with easy navigational capabilities. You can control the display of the checker results using various environment settings.

For more information about the enhanced checker and the checker-related environment settings, refer to the eDeveloper Reference Guide.

Tools Infrastructure

You can utilize the new tools infrastructure, which lets you manipulate your applications using a collection of operations and enhance your development platform by adding user-defined tool applications.

For more information on the Tools infrastructure, refer to the eDeveloper Reference Guide.

The Application Wizard

The Application Wizard is an eDeveloper application that takes advantage of the Tools infrastructure to provide a user-friendly interface for creating various application objects.

To run the application wizard, click the Application wizard option on the Tools menu from an open application.

List Box – Multi-Selection Support

The list box GUI control supports a multiple selection mode.

For more information about the list box multiple selection support, refer to the eDeveloper Reference Guide.

Shared Values Support

Shared values are similar to Global values that are set by SetParam. However, unlike global values, which are only available to the context that created them, shared values are shared and available from all contexts of the same server process.

You can use the SharedValGet and SharedValSet functions to handle Shared values.

For more information about these functions, refer to the eDeveloper Reference Guide.

Pack and Unpack Global Values

You can transmit global values that are set by the SetParam function from one context to another using the ParamsPack and ParamsUnpack functions.

For more information about these functions, refer to the eDeveloper Reference Guide.

Handling Time Values in Milliseconds

You can retrieve and handle time values in milliseconds using three new functions: mTime, mTStr, mTVal.

For more information about these functions, refer to the eDeveloper Reference Guide.

Table Owner – Table Property – Support Logical Name

The Table property of Table Owner can be defined using a logical name that will be translated every time the table is opened. This enhances the ability to define the table's definition using logical names and thereby enables you to dynamically set the table definition at runtime.

For more information about this property, refer to the eDeveloper Reference Guide.

Handling Valid Data for XML Documents

XML documents created by Merge forms must be converted to replace all special and illegal characters to a valid representation of these characters.

You can convert regular strings to valid XML strings and vice versa using the new XMLVal and XMLStr functions.

A new property of the Merge form, XML Output, automatically converts all strings to valid XML data.

For more information about this property and these new functions, refer to the eDeveloper Reference Guide.

eDeveloper 9.4 Service Pack 1

Fixed Problems and Change of Behavior

Browser Client – Task Caching – Current Language Character Set

The Browser client feature using the new Task Caching mechanism requires a modification in the ANSI to Unicode translation environment setting. The character set of the language in use should be set using the Standard name and the Java name. For example, if the language of a browser client application is Korean than the environment setting should include both the “MS949” Java name and the “windows-949” separated by slash.

For example: MS949/windows-949

You may use the following link to find the Standard name and Java name of various character sets:

http://www.devsphere.com/mapping/docs/guide/encodings.html

Server Engines – Graceful Termination

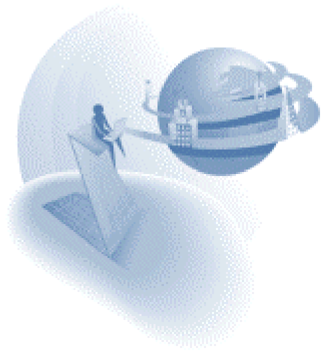
Before version 9.4 SP1, any request to terminate the server engine would abruptly terminate the engine in a way that any running thread was aborted and any kept context was cleared.

Now, when an engine is set to terminate it terminates gracefully, meaning that the engine ceases opening new contexts and only keeps on serving requests of already opened and kept contexts, namely contexts of Browser Client applications. Only after the existing contexts are terminated by a proper exit of the context or by exceeding the context inactivity timeout, does the server engine terminate.

You may set a timeout value for the termination procedure and instruct the engine to terminate abruptly if there are still opened contexts after the set timeout.

The timeout can be set for every terminate operation in each of the following ways:

- RqRtTrmEx function – An extension of the RqRtTrm function that supports an additional parameter of the timeout.
- Command line requester – Sets the timeout using the –Timeout= parameter.
- SNMP – In SNMP monitors that support single parameters, you may set the timeout in the first parameter after a preceding comma delimiter, such as host/port,timeout. In other monitors, the second parameter of the Terminate operation is the termination timeout.



Messaging Connectivity

In order to dynamically define the place of the messaging component and its configuration files, two new logical names have been added:

MessagingComponentDir - This points to the place of the messaging component. It must end in a slash, depending on the Operating System. For Windows this is \.

MessagingErrorLogFile - This is the name of the error log. The file will be placed in the MessagingComponentDir

Some new errors have been added. New functionality has been added to the Error List screen, which lets the developer export and import the error list. If you make any changes to the error text, you can save the changes using the export / import functionality.

Tree Control Images

The Expanded Image Index and Collapsed Image Index properties now serve as the default image of the Expanded and Collapsed states of the tree node, unlike in the previous version where these images were only used for the non-parked state. This means that from this version on, if an image index is set for the Expanded Image Index property but no image is set for the Parked Expanded Image Index, the same image will be displayed regardless of whether or not the Tree node is parked on.

Language Translation of Internal Message Boxes

Strings of internal message boxes, such as the Confirm Delete operation, were not translated by the Multi-Language Support mechanism.

Trim Function (QCR 4350)

The Trim function behavior was slightly modified in version 9 so that null characters at the end of the string were trimmed. This new behavior is now only available by setting a special flag as follows:

```
[MAGIC_SPECIALS]
SpecialTrimTrailingNulls=Y
```

In the default behavior of the Trim function trailing, null characters are not trimmed.

Numeric Variables of a Left-Aligned Picture (QCR 450461)

In previous versions, dropping a numeric variable of a left-aligned picture kept the Horizontal alignment property of the edit control inherited with a value of Right, which contradicted the field's picture. Starting from this version, dropping a numeric variable of a left-aligned picture automatically breaks the Horizontal alignment property of the control to the Left value.

Table Row Highlighting (QCR 138940)

In previous versions the row highlighting of the current record was set only after the completion of the Record Prefix handler of that record. This behavior was fixed, and now the row highlighting is set before the Record Prefix.

Password and Multi Line properties of Browser Client Edit control

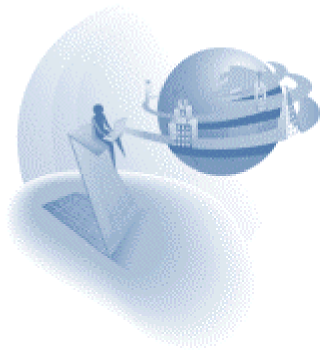
The Password and Multi-Line properties of a browser client edit control only affect the control when the control is placed on the form. Therefore, these properties are now only available at the field level and are always disabled in the property sheet of the actual HTML control.

eDeveloper Error Log

The eDeveloper error log used to be created as MGFLWMTR.LOG. Now the error log file name is MGERROR.LOG

Tree Control Related Events Names – Hebrew Language

The Hebrew translation of several tree control events has been changed. This is not noticeable when a 9.4 application is directly opened by a 9.4 SP1 engine. Importing a 9.4 Hebrew application to 9.4 SP1 may loose references to these events.



Features and Modifications

Browser Client – Task Caching

The browser client technology of eDeveloper has gone through a major performance enhancement by enabling the end-user’s browser to locally cache the logic segments of the application. In this way a significant amount of data is retrieved from the client’s cache instead of being transmitted anew for every task. The browser client caching mechanism is aware of changes introduced to existing tasks and will cache and utilize the modified task. This mechanism is affected by three new environment settings: Two new environment properties under the Server tab of the Setting\Environment dialog: Browser client cache path and Browser client cache alias and an additional, application specific setting of Revision.

For more information on the environment settings, refer to the eDeveloper Reference Guide.

Flow Monitor – Dynamic Append of Variables

You can set your application to dynamically append the variables list to the flow monitor log at runtime using the new FlwMtrVars function.

For more information on the FlwMtrVars function, refer to the eDeveloper Reference Guide.

Variable IO – Supporting Line Format

The Media type option for Variable introduced in 9.3 SP4 only supported the None format mode. The Media type option for Variable now also supports the Line format option.

For more information on the IO media type Variable, refer to the eDeveloper Reference Guide.

IN function

The IN function is now available to be used either as part of your regular application logic and as part of the Magic Where clause of a task that handles an SQL table. Using the Magic IN function in the Magic Where Clause property will be automatically translated to the expected syntax of the underlying DBMS.

For more information on the IN function, refer to the eDeveloper Reference Guide.

eDeveloper Components – Execution Indication

An application can sometimes be executed as the main application and sometimes as a component of another application. The logic of the application may vary according to its execution.

A new IsComponent function lets you identify whether an application is executed as a component application or as the main application.

For more information on the IsComponent function, refer to the eDeveloper Reference Guide.

Cross-Contexts Event Posting

The Raise Event operation was enhanced to enable posting an event from one context to another. This can be done only on a background enterprise where several contexts may be handled by the same process.

Posting an event from one context to another can be done by first naming the target context using the CtxSetName in the target context, and then raising the event using the Raise event operation by specifying the name given to the target context.

For more information on the CtxSetName function and the enhanced Raise Event operation, refer to the eDeveloper Reference Guide.

Variable IO - XML Functions support

The XML function used to handle an XML document defined only as a designated IO of an XML Direct Access media type. The XML Direct Access media type can only refer to physical XML files.

Now you can use the XML functions to handle XML documents defined by a Variable media type. The Variable media type lets you keep an XML document as a variable in the memory.

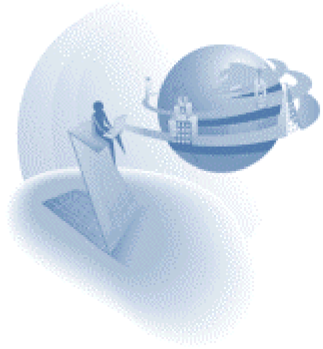
For more information on the XML-related functions and the Variable IO, refer to the eDeveloper Reference Guide.

Order By Left Outer Join Variables

Sort settings of a task that its segments are variables of a left outer join link operation will produce a proper Order by clause in the result select statement.

Expression Editor's Program List – Arguments View

You can browse through the expected arguments of a program by zooming in from the program list available in the expression editor dialog.



UTF8 Support

You can convert eDeveloper ANSI strings to UTF8 and vice versa using two new functions: UTF8ToAnsi and UTF8FromAnsi.

For more information on these functions, refer to the eDeveloper Reference Guide.

Tree Control – Expand and Collapse Node Events

The Expand and Collapse node events may be triggered on tree nodes that are not parked on. Therefore, retrieval of the tree node value and the tree node level of the expanded\collapsed node cannot be done using the TreeLevel and Tree Value functions.

These values are now transmitted as arguments of the Expand Node and Collapse Node events. When creating a handler on such an event, eDeveloper suggests that you create two virtual variables. These variables are updated with the node value and level when the Expand\Collapse event is triggered.

Tree Control – Parked Images

In the previous version, you could only specify a single image for the parked tree node, regardless of the tree node expanded/collapsed state.

In this version the parked image has been split into two separate images:

- Parked expanded image index
- Parked collapsed image index

This way you can set different images for every state.

Tree Control – Edit Node Event

The edit action of a tree node is now available using a new Edit Node internal event. In modify mode, a secondary left click on a tree node triggers the event and causes the tree node to enter edit mode. You can switch the tree node into edit mode by directly triggering the Edit Node event.

Deferred Transaction Cache Logging

A new Flow Monitor activity filter, Transaction Cache, lets you view the records' manipulation instructions of a deferred transaction.

To view the transaction cache, just activate the Transaction Cache activity filter in the flow monitor.

COM Objects – Retrieval of Two-Dimensional Arrays

A single dimensional array value to be retrieved from a COM object can be retrieved using a Vector field. The cell model of the vector field should match the data type of the array.

From this version on, you can also retrieve two-dimensional arrays using a vector field whose cell model is also a vector.

Overwriting a Table Entry – Advanced Mode

The table object of an application in the Table repository differs from other application objects in that its internal elements, columns and indexes, can be referenced from other objects in the application.

When overwriting a table entry with another table entry, by default, the column and index internal references are kept according to the existing references of the overwritten table following their sequential order.

A new Advanced button on the Overwrite Table dialog shows advanced options to finely tune the overwrite procedure.

Direct SQL – Binding Variables Support

You can define Direct SQL statements by binding variables. This options enables you to improve the performance of such statements.

For more information on Binding variables, refer to the eDeveloper Reference Guide.

Web Services – Header Information

Header information of a Web Service call can be retrieved as an argument of the call operation. The Web Service Assist mechanism creates a corresponding argument entry for the header information.

Blob Columns – Default Mapping for Oracle

Blob columns in Oracle tables are mapped by eDeveloper to the LONGRAW type.

The LONGRAW type has some restrictions that are no longer encountered when the columns are mapped to the BLOB type.

Instead of explicitly setting the Type property of every blob column to BLOB, you may instruct the Oracle gateway to automatically map every blob column to the BLOB type by placing the following flag in the Database Information property of the Oracle database entry:

```
Default_Blob_to_Blob=Y
```

Foreground Generator Context Management

In previous versions, the foreground generator engine served requests in a single mode similar to the way a runtime foreground engine handles requests. The generator could not handle requests in the same way they could be handled in deployment by a background engine.

A new environment setting, ***Foreground Generator Context Management***, located on the Server tab in the Environment settings dialog, lets you set the mode in which the foreground generator engine handles requests. The two available modes are:

- ❑ As background engine
- ❑ Single common context

When this setting is set to *As background engine*:

- ❑ Upon switching to Runtime mode the Task Prefix of the Main Program is not executed.
- ❑ The Menu system displays a built-in menu structure. The user-defined pulldown menu is not displayed.
- ❑ Main Program events are not activated at idle time.
- ❑ For every explicit request:
    - o The Task Prefix of the Main Program is executed
    - o A new context is created
    - o When the request is completed or, in the case of a Browser task, when the top-level task is exited, the Task Suffix of the Main Program is executed and the context is cleared.

Opening a new context for every explicit request means that:

- ❑ All Main Program variables are reset
- ❑ All memory tables are reset
- ❑ All Global Values are reset
- ❑ The Environment settings are reset.

The *As background engine* mode described above is the way requests are handled by a background engine.

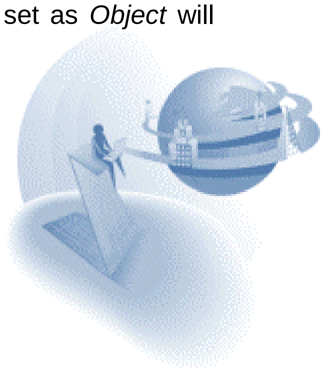
Important: The new *As background engine* mode is the default mode, therefore causing a change of behavior. Changing the value to *Single common context* results in the previous behavior.

COM Objects – Removal of Stored as Property

The Stored as property of the OLE and Active-X fields, in conjunction with the Instantiation property, has become redundant and has been removed in service pack 2.

Objects that had their ***Stored as*** property set as *Reference* will have their Instantiation property set to None.

The Instantiation property of objects that had their ***Stored as*** property set as *Object* will keep their previous value.



MTblSet Function – Syntax Modification

The syntax of the MTblSet function has been modified to support an additional parameter.

The function of the previous syntax was deprecated and has been renamed MTblSet1. Any use of the MTblSet function in a previous version will be regarded as using the deprecated MTblSet1 function. We recommended replacing any MTblSet1 usage with the new MTblSet function.

Important: Replacing the previous MTblSet function with the MTblSet1 function is only done when opening an application that was developed in eDeveloper 9.4 before Service Pack 2 directly in Service Pack 2. However, when you import an export file from such an application, expressions that use the MTblSet function in its previous syntax will result in incorrect expressions. You should either open such applications directly in Service Pack 2 or simply open the export file and replace any occurrence of the MTblSet string with the MTblSet1 string.

For more information on the MTblSet function, refer to the *eDeveloper Reference Guide*.

DropMouseX & DropMouseY Function – Return Value Modification

The DropMouseX and the DropMouseY functions used to return the coordinates of the mouse cursor in pixels

Starting from Service Pack 2, the functions return the mouse cursor coordinates in the unit of measurement used by the current form.

For more information on the DropMouseX and DropMouseY functions, refer to the *eDeveloper Reference Guide*.

Choice Controls – Renaming the Label Property

The ***Label*** property of choice controls has been renamed as ***Items list***.

Choice Controls – Handling of Last Single Backslash Character

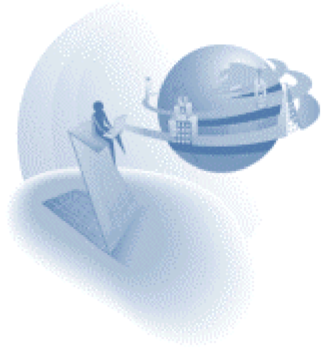
The Backslash character instructs the engine to handle the character that follows the backslash as a regular character and not as an operator. For example, this can indicate that a comma should be regarded as a simple character and not as a delimiter.

The last single backslash character is regarded as a backslash character followed by an empty space, which results in displaying an empty space.

This is a change of erroneous behavior in previous versions where the last single backslash character resulted in displaying a backlash character.

Both in previous versions and in Service Pack 2 displaying a backslash as the last character of the last item is done by placing two backslash characters.

For example: A combo box that has the items of *A:\* and *C:\* should have its items list set with *A:\\,C:\\*.



Multi-Line Edit Control – Modified Edit Length (QCR 780336)

A Multi-Line Edit control used to enable text entry according to the length of the attached variable and not of the actual picture that was set for the control.

Event Handlers – Evaluating Expression (QCR 441551)

A zero task generation used in a function, such as Level(0), that is evaluated by an ancestor task handler used to refer to the task where the handler is defined and not to the task that triggered the event.

Jump to Row in a Folder (QCR 757811)

The Jump to Row utility, executed by the Ctrl+J key combination, now refers to the given row number as the fixed row number of the object and not as the relative row within the viewed folder as in previous versions.

Browser Client – ANSI to Unicode Translation – Default

Since eDeveloper 9.4 Service Pack 1, it is required to state proper code page values in the ANSI to Unicode Translation environment setting.

eDeveloper 9.4 Service Pack 2 provides a default value for this setting according to the language of the Mgconstw file.

This means that if no value is set for this environment setting, which is the installation default setting, a proper value will be taken from the Mgconstw file.

Note: If you wish to have no value set for the encoding of the page, enter the string **None** in the ANSI to Unicode Translation.

WebSphere Messaging Component Changes

A parameter was added to the Send and Quick Send programs. This optional parameter informs the messaging server what extra actions should be performed on the message. For more information, read *eDeveloper's Messaging Connectivity Component* guide.

New Features

Split Window

The GUI Display form lets you split the window to display two forms of two tasks. The splitter that divides the window lets the end-user change the proportions of each window part. You can define a vertical or horizontal splitter, and you can also define nested split windows.

For more information on split window properties, refer to the *eDeveloper Reference Guide*.

Choice Controls – Returned Values and Displayed Values

In previous versions, the displayed value of a range-based choice control was always identical to a corresponding returned value.

Using the new ***Display list*** property, you can now set a choice control with a displayed items list. Selecting a choice-control item returns the corresponding value defined by the ***Items list*** property, formerly named ***Label***.

For more information on the Items list and the Displayed list properties, refer to the *eDeveloper Reference Guide*.

Data Controls – Additional Values

You can define the ***Source table*** related properties together with the ***Items list*** and ***Display list*** properties to create a group of options taken from the source table’s records and a collection of simple items.

This feature lets you add general options such as All, [Blank], None options to a data-bound choice control easily.

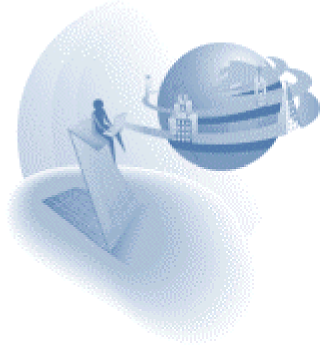
For more information on the Items list and the Displayed list properties, refer to the *eDeveloper Reference Guide*.

Context ID – Retrieval by Name

The new ***CtxGetID*** function lets you retrieve the ID of a current context or another context by its name.

Note: A context can be named by the ***CtxSetName*** function.

For more information on the CtxGetID and the CtxSetName functions, refer to the *eDeveloper Reference Guide*.



COM Objects – DCOM Support Enhancements

In previous versions, the only way a COM object could be instantiated as a DCOM object was by configuring the object run on a remote machine as part of the general setting of the object for the entire client machine. This way the application did not have a way to explicitly

define the exact remote host machine where the object should be executed as a DCOM object.

Starting from Service Pack 2, you can instruct a COM object to be executed as a DCOM object on a specific machine using the new ***Remote host*** property of the OLE field.

For more information on the Remote host property, refer to the *eDeveloper Reference Guide*.

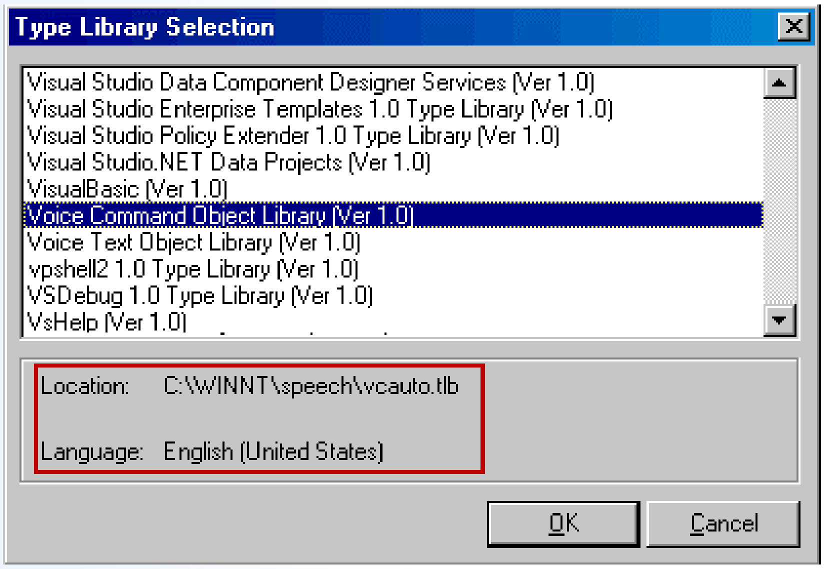
COM Objects – Non-Sequential Collections Support

Some COM objects include Collection type sub-objects. A collection item is usually retrieved by an ID field. Sometimes the ID of a collection item is not a sequential identifier; and there is therefore no direct way to retrieve items sequentially.

Starting from Service Pack 2, you will notice an additional method for every collection sub-object. This method, named ***MGItemSequential,*** is an internal method for eDeveloper clients that lets you retrieve an item from the collection by specifying its sequential index.

COM Objects – Object File Location

The Selection list of an OLE object or an Active-X control displays the file location and language of the marked object



Visual Studio Data Component Designer Services (Ver 1.0)
Visual Studio Enterprise Templates 1.0 Type Library (Ver 1.0)
Visual Studio Policy Extender 1.0 Type Library (Ver 1.0)
Visual Studio.NET Data Projects (Ver 1.0)
VisualBasic (Ver 1.0)
Voice Command Object Library (Ver 1.0)
Voice Text Object Library (Ver 1.0)
vpshell2 1.0 Type Library (Ver 1.0)
VSDebug 1.0 Type Library (Ver 1.0)
VsHelp (Ver 1.0)

Location: C:\WINNT\speech\vcauto.tlb

Language: English (United States)

Figure 1: The bottom section of the selection list displays the file location and language of the object.

Broker & Requests Monitor Utility

The eDeveloper 9.4 Service Pack 2 provides a new utility that monitors and handles a selected broker and its requests.

The new utility lets you examine various types of information about the broker and its requests in concurrent panes.

You can run the monitor utility by executing the MGRQMonitor.exe file in your eDeveloper directory, or from the context menu of the Broker icon on the System Tray, as shown in figure 2.

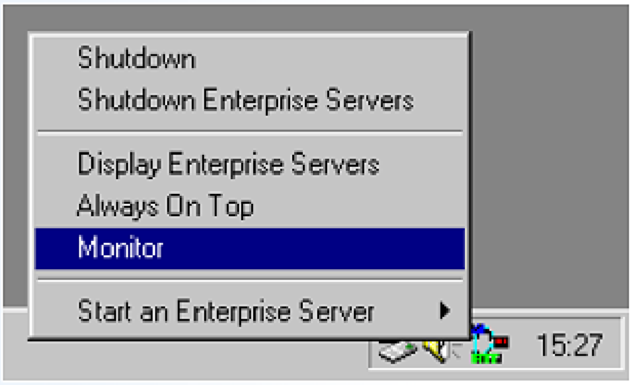

Shutdown
Shutdown Enterprise Servers
Display Enterprise Servers
Always On Top
Monitor
Start an Enterprise Server
15:27

Figure 2: The Context Menu of the Broker showing the option to run the Monitor

Deferred Transaction – Support Direct SQL Statements

Deferred transactions can be defined for tasks that are set by a Direct SQL statement.

Note: Table-based tasks in a deferred transaction reflect modifications of the tables that were entered by another task within the same transaction. However, a Direct SQL-based task does not reflect such modifications.

HTTP Authentication Support

eDeveloper supports web server authentication required for any HTTP reference performed by the engine. HTTP references can be in runtime using the HTTPPOST and HTTPGET functions, and using the Web Service call operation, or in toolkit using the WSDL assist utility of the Web service call operation.

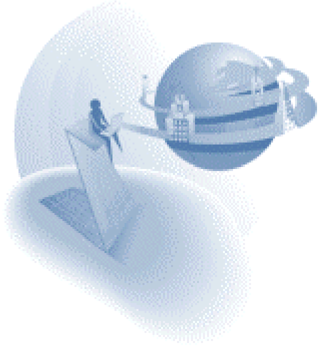
The username and password that need to be passed along the URL should be added as part of the URL in the following manner:

```
[Protocol]://[User]:[Password]@[rest of the URL]
```

For example: `http://John:MyPass@WWW.MySite.COM/`

The HTTP Authentication mechanism supports both Basic and Digest authentication modes.

You can use secret names at in any part of the URL.



Web Service Attachments

When executed as a web-services provider, eDeveloper supports DIME attachments for both RPC and DOC/Literal. This functionality addresses the requirement of embedding binary data into web services.

Using two new functions, WSAttachmentGet and WSAttachmentAdd, a web-service provider program can retrieve attachments from, or add attachments to, the SOAP response.

For more information about these functions refer to the *eDeveloper Reference Guide*.

Codepage Setting for XML Documents

Using the CodePage function, you can set the codepage of an XML document handled by your application to be different from the operating system codepage. The CodePage function is effective within a specific running context.

For more information about this function refer to the *eDeveloper Reference Guide*.

eDeveloper 9.4 Service Pack 2a

Fixed Problems

Group Level Updates (QCR 999112)

Group level updates were not performed properly if the task included a BLOB virtual variable.

Group Control Display (QCR 440406)

A group control set with no title text was not displayed properly.

HTTP Variables Retrieval (QCR 983833)

The engine could not retrieve the value of an HTTP variable that was defined as the first HTTP variable in the Mgreq.ini file.

Range on Link Operation Return Code (QCR 293169)

A data view range was not performed properly if it was defined on the return code of a link operation.

eDeveloper 9.4 Service Pack 2b

Fixed Problems and New Features

QCR 994543

When ranging on Logical fields, the "To" range value was not updated with the "From" range value, during a user defined range.

QCR 835294

Wrong date appeared when displaying the DSTR value of a date after emptying it, providing the date is null allowed, its null value is 00/00/0000 and its null default is set to Yes.

QCR 979594

A flicker was caused by redundant repaint of an OCX sending events to the engine.

Messaging Component – MQ support

The following functionalities have been added:

- Publish and subscribe.
- Buffer management – Two new programs: One that adds a single string to a buffer and another that fetches the string from the buffer.
- Different options for disconnecting from a queue.
- Activity log can be activated with a function and by logical name.
- When subscribing and un-subscribing the information is logged to the MQ_Subscriptions.log

eDeveloper 9.4 Service Pack 3

Important Notes

eDeveloper 9.4 Samples Package

The Samples package demonstrating the new eDeveloper 9.4 features has been updated with additional samples. You can download the updated Samples package from our FTP site using the following link:

ftp://v9product:PrEver983@ftp.magicsoftware.com/v9product/94samples/

Tools Infrastructure Sample & Guide

This release provides you with a sample and guide describing how you can create your own add-on tools for your development environment.

You can download the ToolSample.exe package from the Samples section of our FTP site.

.NET and eDeveloper COM Interoperability Sample & Guide

Download the DotNetInterop.exe package from the Samples section of our FTP site to learn how to access .NET assemblies from any eDeveloper application, using .NET COM interoperability.

Phantom Task – New Term

This document and the SP3 Samples package both refer to a new eDeveloper term: “phantom tasks”. A phantom task is a task that although closed, keeps its window open (Close Task Window=No).

Problems Fixed and Behavior Changes

Call UDP - Change Convention Default

Most calls of Windows Operating System DLL files require the use of a Standard call convention. For this purpose, the default value for the Call DLL operation convention property is now set to Standard.

This modification only affects newly-created Call operations and has no effect on existing Call operations.

Main Program Windows - Click Effect

Clicking a displayed Main Program window used to close any running task. This behavior has been fixed and click events on the Main Program form are now handled in the same way as clicks on the MDI client area.

An exception to this rule is “push buttons”. Clicking a “push button” that is placed on the form will close all active tasks.

EJB Builder - Package Name Change

The package name of an EJB generated with eDeveloper has been changed to com.magicsoftware.j2ee.

The previous package name, magic.j2ee, is still supported so that EJBs generated by previous versions of eDeveloper 9.4 will still work.

CTXKILL and CTXCLOSE Functions

The CTXKill function has been renamed “CTXClose” to indicate the function’s true behavior of gracefully closing a context.

A new CTXKill function has been added, which gives you the ability to abruptly close a context.

Important: When you open or import an existing application based on a previous 9.4 release, the existing function will automatically display as CTXClose.

WMQ - Fixes and Supplements

Buffer Conversions

The following new functions were added for easy handling of text-to-buffer and buffer-to-text routines:

- GetTextFromBuffer
- SetTextIntoBuffer

Log Enhancements

The following two additional log files are now supported:

- Component Activity log
- Subscribe/Unsubscribe log

For more information on WMQ, see the *eDeveloper's Messaging Connectivity Component* guide.

MLS Effect on Data Controls

The eDeveloper Language Support mechanism used to translate a data control's item collections. This behavior has been corrected in SP3 so that data control items fetched from the data control table are not subject to the MLS mechanism.

In the past, the MLS mechanism did not translate the items set by the Items list and the Display list properties. Items set by these properties are now handled by the MLS mechanism.

eDeveloper Builders: Setting the Builder Data File Locations

A new section in the Magic.ini file lets you define the location where the data files of each builder are saved. The new section is [Interface_Builders].

Each builder has its own setting within the new section as described below:

- COM_DATA= (COM Interface Builder)
- WS_DATA= (Web Service Interface Builder)
- JCG_DATA= (Java Component Generator)
- XCG_DATA= (XML Component Generator)
- MCI_DATA= (Magic Interface Builder)
- EJB_DATA= (EJB Interface Builder)

For example:

```
[Interface_Builders]
COM_DATA = D:\940\Components\com\
MCI_DATA = %MyDrive%\MCIBuilders\
```

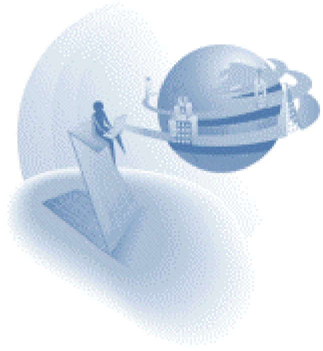


Image Caching: No Cache Setting (QCR 979900)

You can configure the engine so that it does not keep a cache for displayed images. You do this by entering –1 in the **Image Cache Size** Environment setting (Settings/Environment/Preferences).

GUI Display Form: Split Section

The GUI Display form's Split section has been moved above the Navigation section.

Radio Button: Single Option Support

You can now configure a Radio button control with just a single item even if it is linked to a source table.

The radio button supports a single item by setting the following special flag:

```
[MAGIC_SPECIALS]
SpecialSingleItemRadio=Y
```

Resizing Phantom Tasks

Resizing phantom tasks now makes the engine switch to the resized task.

Flickering Interface in Terminal Clients

In some cases the engine's interface would flicker indefinitely on terminal clients. You can use a new Specials flag to solve this problem, as shown below:

```
[MAGIC_SPECIALS]
SpecialPaletteChangeInfiniteLoop=N
```

MLS on CtrlName Function

The Language Support mechanism used to affect the result of the CtrlName function. This problem has been corrected.

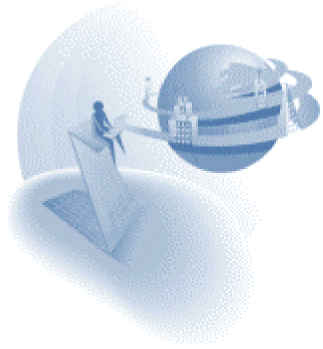
Font Size Greater than 99 (QCR 977849)

In previous versions you could not set a font size greater than 99. You can now enter greater values.

DB2 Gateway - Behavior Change

Due to a behavior change in DB2 V8, the default value of SQL_EXTENDED_FETCH has been changed to N. If you want to maintain the previous behavior, you should enter the following value in the DB2 Database's Database Information property in the Magic.ini file:

```
SQL_EXTENDED_FETCH=Y
```



New Features and Feature Enhancements

Browser Client: .Net Support

eDeveloper 9.4 SP3 supports a .NET-based Browser Client.

This module requires the Microsoft® .NET Framework Re-distributable Package to be installed on machines running the Browser Client. This framework is already an integral part of the latest operating systems provided by Microsoft®, and you can download it from the following location:

http://www.microsoft.com/downloads/details.aspx?FamilyID=262d25e3-f589-4842-8157-034d1e7cf3a3&DisplayLang=en

eDeveloper 9.4 will continue to include the Java-based Browser Client, but this client will not be enhanced. Fixes to the Java-based Browser Client will be decided on a case-by-case basis.

You can configure the enterprise server to execute Browser Client applications using one of the following client-side technologies:

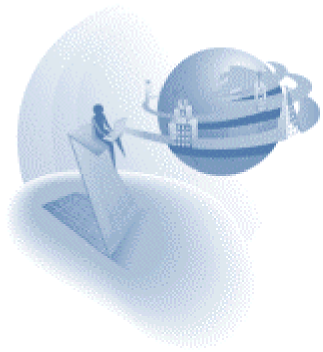
- Microsoft .NET
- Microsoft JVM

For more information on the Browser Client configuration, see the new ***Browser Client Technology*** environment setting (Settings/Environment/Server).

Context Management for Batch-based Requests

Browser Client technology introduced a powerful mechanism for full and automated context management. Until SP3 this mechanism could only serve Browser tasks. From SP3 onwards, you can also use this mechanism for batch-based requests, i.e. merge-based requests.

For more information about this and to learn how to use this new feature see the Samples Package of this release.



Browser Client: Java-based Module Installation

When you use the Microsoft JVM as the infrastructure for Browser Client applications you need to have two modules installed on the client side. These modules are automatically installed the first time you access the URL of a Browser Client application. For installation purposes, the client must be logged on as a user with administrator privileges.

Clients that are not usually logged on with administrator privileges can only execute Browser Client applications if an administrator explicitly installs the modules on that client.

Using the BCJavaInstall.exe utility, located under the Support folder of you eDeveloper installation, you can install the Browser Client Java modules on a client machine, using an administrator user for the installation sequence.

Syntax: BCJavaInstall <URL> [<*Username*> <*Password*>]

Where: **URL** is a URL to a valid virtual directory where the COM object (MGBCCOM9.cab) and the persistent applet (MGBCxxx_yyS.cab) reside. For example: http://servername/Magic94Scripts/

Username is an optional parameter. If you provide this parameter, the installation is carried out for this user. You should use this option if the currently logged user does not have the appropriate privileges for installing the COM object and persistent applet.

Password is an optional parameter that holds the user's password. You should only specify this parameter if you also specified the username parameter.

Note: This utility only supports lowercase passwords. The username and password parameters are not supported on Windows 9X. If the password you want to use is either of uppercase or mixed case, or if you want to install the modules on Windows 9X, you should log on to the operating system with a username and password having appropriate privileges.

An installation utility for the Browser Client .NET modules will be provided at a later stage.

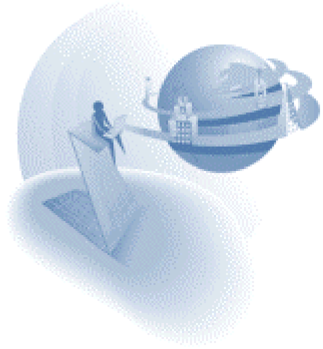
XML Direct Access: Write Functions

eDeveloper lets you modify an XML document using direct access by providing these new XML functions:

- ❑ XMLInsert
- ❑ XMLModify
- ❑ XMLDelete

For more information on these new functions, see the *eDeveloper Reference Guide*.

To learn how to use these functions, see the Samples Package of this release.



XML Namespace Support

XML Direct Access support has been enhanced to support usage of XML namespaces and aliases. All the XML direct access functions are affected by this enhancement.

You can use the functions in two modes:

- Namespace-aware mode
- Non-namespace aware mode (as the functions behaved until this version)

The enhancement includes two new functions:

- XMLSetNS
- XMLGetAlias

For more information, see the *eDeveloper Reference Guide*.

Secured HTTP Communication with Client Certificates

To provide more secure HTTPS (HTTP over SSL) communication, some Web sites require the client to provide a certificate that identifies itself to the server as part of the initiation of the secured communication. This SSL client certificate should be issued by a trusted organization and should identify the holder of the certificate.

The HTTPGet and HTTPPost functions, as well as the Call Web Services operation (Call Web.S), have been enhanced. You can now configure eDeveloper to send a client certificate to the Web server that receives the HTTPS call.

The following new functions were also added:

- ClientCertificateAdd
- ClientCertificateDiscard

The following new settings have been added to the Settings/Environment’s External tab:

- SSL CA Certificate files
- SSL client certificate file
- SSL client certificate password

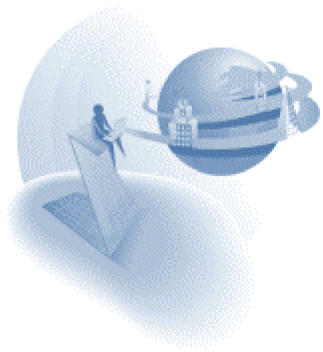
For more information, see the *eDeveloper Reference Guide*.

Drag-and-Drop: File Format Support

Drag-and-Drop functionality has been enhanced to include support for the file format. This enhancement lets you get the file name and path of the file dragged from Windows Explorer.

You can see an example of this enhancement in the 9.4 SP3 Samples package.

For more information about eDeveloper’s Drag-and-Drop support, see the *eDeveloper Reference Guide*.



Phantom Tasks: Re-entrance Through Handlers

In previous versions you could only return to a phantom task by a Click event, if a corresponding Call operation was set in the parent task's Record Main.

The runtime engine has been enhanced to enable re-entry to phantom tasks by searching for a corresponding Call operation in almost any handler, and not just in the Record Main.

You can see an example of this enhancement in the 9.4 SP3 Samples package.

Tools Infrastructure Enhancements

The Tools Infrastructure has been enhanced with two additional capabilities:

Support Export of Parked Item

You can configure a tool application to only export the object on which the developer is parked when activating the application. You can enable this by using the ***Range=@*** switch in the export operation line.

Support Overwrite on Import

You can instruct the Toolkit engine to import an object and have the newly imported object replace an existing one, by using the new ***OverwriteTarget*** switch.

You can see an example of these enhancements and learn more about the Tools concept in the Tools Sample.

Identify Modified Row for ISAM Tables

The Identify Modified Row capability, once available for SQL tables only, is now also available for ISAM tables.

For more information on the Identify Modified Row capability, see the *eDeveloper Reference Guide*.

UserDel Function

The new UserDel function is a supplement to the existing UserAdd function. The UserDel function lets you delete users from the users file at runtime.

For more information about this function, see the *eDeveloper Reference Guide*.

XSD to be Added to Print Data Output

The Print Data utility's XML output has been enhanced to create an XSD that corresponds with the generated XML file.

XCG: Interface Changes and Namespace Support

The XML Component Generator (XCG) is now in wizard format.

If you launch the XCG from the New Application dialog box, the resulting component will automatically be imported to the new application.

The XCG now also supports namespaces.

For more information about the XCG, see the *eDeveloper Reference Guide*.

eDeveloper 9.4 Service Pack 3a

Problems Fixed and Behavior Changes

Table Control in Window – Placement Calculation (QCR #98)

Placement was not properly calculated when a table control is set to be a Table in window.

.NET Browser Client Served by Apache Web Server– (QCR #1020)

When using .NET Browser Client served by Apache Web Server, the required MSHTML module (MshtmlSetup.msi) was not installed automatically.

Invisible Cursor in an Edit Control (QCR #296085)

The cursor was not shown in an edit control after calling and immediately closing a child task.

Mail Functions – Space Characters in Mail Subject (QCR #741224)

Space characters were replaced with underscore characters in the mail subject when the mail subject contains also non-English characters.

Form Editor Crash in a Subtask (QCR #741349)

eDeveloper crashed on exiting the form editor of a subtask having the form of its parent task set to be a child form.

Crash on Export Document of Menu Repository (QCR #742322)

eDeveloper crashed on exporting the Menu repository when the *Foreground Generator Context Management* environment setting is set to *Background*.

COM Support – Passing Data Types (QCR #743540)

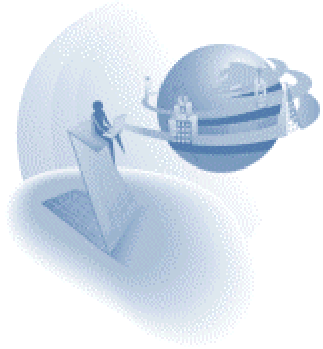
In specific scenarios wrong data types were passed to a COM object.

Memory Leak in Resident Tables (QCR #915803)

A memory leak was noted on loading a resident table in conjunction with task sort.

New XML Component (QCR #922180)

A new application designed to be an XML component remained empty after completing the XCG wizard flow.



Team Development & Components on iSeries (QCR #973272)

eDeveloper crashed on checking-in models of an application used as a component when the application is on iSeries.

COM Support – Passing Objects (QCR #996176)

Passing an object variable as an argument to another task was not updated with the handle of the called task parameter if the instantiation of the object variables was set to None.

J2EE Support (QCR #997253)

Wrong binary data was passed to an eDeveloper program with a BLOB parameter when calling an eDeveloper generated EJB (Blob mapped to Java byte array).

eDeveloper 9.4 Service Pack 3b

New Features and Behavior Changes

UDDI Utility

The new eDeveloper UDDI utility lets you search a Web Service on a UDDI server and automatically creates a program that calls the selected Web Service. The UDDI utility also lets you publish your Web Service on a UDDI server.

For more information, see the Utilities chapter of the *eDeveloper Reference Guide*.

Automatic Commit of Oracle Select Statements Removed

Select statements in an Oracle-based task without transactions were set with an automatic execution mode of Commit. This resulted in an inflated log file containing many redundant entries. In SP3b, the execution mode for these Oracle Select statements is now set to OCI_DEFAULT.

Focus on Right-Click

Right-clicking on a form or on a control of any task, whether the current task, ancestor task, or phantom task, did not set the focus to the form or control as should have happened.

Mgreq.ini Search Sequence

In previous versions, the eDeveloper engine loaded the Mgreq.ini file that exists in the engine's activation working directory. If the file was not found there, the file was not loaded and default values were used. In SP3b, this behavior has been changed, so that if the Mgreq.ini file is not found in the working directory, the engine tries to load the file from the directory from where the messaging layer module (Mgrqgnrc.dll) was loaded.

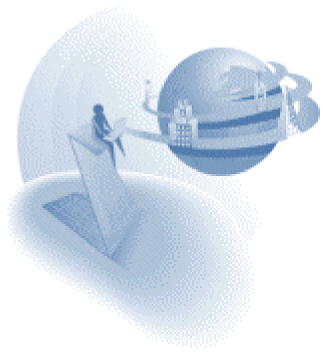
.NET Browser Client - Performance Enhancement

The loading of a page on a .NET browser client application was improved by evaluating the initial value of the Visible and Enable expressions of every control already on the server side. This may cause a change of behavior in controls set with an expression on the Visible or Enable properties and also default values for these properties that are set directly on the HTML page. It is recommended, in such cases, to remove the Visible and Enable default values from the page.

CountersEqualizationDuringStatus – Change of Syntax

Modifications have been made to the syntax of the **CountersEqualizationDuringStatus** broker entry responsible for keeping the broker synchronized with the state of each engine and each request.

In previous versions, this setting was set by either Yes or No. In SP3b, this setting requires a numeric value that determines the time interval in seconds in which the engines will report to the broker on the state of its requests. When set to zero, there will be no synchronization. The default value for this setting is **1** second.



Fixed Problems

Licensing

Various licensing-related problems have been resolved in SP3b.

Proxy Server

Various Proxy-related problems have been addressed in SP3b.

Lock Error on DB2 (QCR # 3251)

When encountering a lock while fetching records in a task with a program sort, the engine continued to prepare the dataview instead of immediately aborting.

Lock Error on DB2 (QCR # 5573)

eDeveloper crashed on evaluating the XMLBlobGet function on an empty element.

QUOTED_IDENTIFIER Flag for MSSQL (QCR # 205206)

The QUOTED_IDENTIFIER database flag did not work properly.

Event Handling on Browser Client (QCR # 289920)

Component events were not properly propagated in a browser client application.

Direct SQL on DB2 (QCR # 291650)

A valid direct SQL command occasionally failed.

Dynamic Column Width (QCR # 295166)

A column defined with a width expression was re-computed without taking account of placement modifications of the form. This resulted in the column width resetting its value when a re-compute cycle begun.

Toolkit Folders (QCR # 305203)

The toolkit engine crashed when clicking the Navigator pane after creating a folder.

Oracle’s CLOB Support (QCR # 698035)

An eDeveloper BLOB column could not be mapped to an Oracle CLOB data type. You can now do this in SP3b. To use this data mapping, you should enter CLOB in the Type property of a BLOB variable. The recommended method of inputting data into a CLOB field is by using the Update operation or by using the File2Blb function.

Large MSSQL views (QCR # 730249)

Large MSSQL views were improperly imported to an application.

Event Handling on .Net Browser Client (QCR # 738318)

In specific cases, the Record Suffix was executed unnecessarily.

Components on a Browser Client (QCR # 743314)

Some components that handled events of other components were not always loaded properly and caused an “*In the parsing”* error on the page.

Requests Monitor (QCR # 743552)

The user name of a context was not shown properly in the Requests monitor.

Reversed Sort on Oracle (QCR # 745762)

When performing a sort on natural data types, data was shown in reverse order after activating the End Table event.

Virtual Variable Update in Query Mode (QCR # 751171)

The “*Warning - Modify operation canceled due to task Query mode*” message was displayed for no reason when updating a virtual variable within a Link Block in a Query mode task.

Jump to Row Event (QCR # 762161)

The Jump to Row event did not work properly after it was performed within the Programs list.

Background Engine (QCR # 764038)

An incorrect result for concurrent requests was recorded when using a background engine.

MCF on Oracle (QCR # 764304)

The engine crashed when saving an MCF as MFF if the MCF resided on an Oracle database and the MCF was in Team Development mode.

Incremental Search on DSQL (QCR # 766180)

Incremental search did not work on a Direct SQL result.

Block Operation on .Net Browser Client (QCR # 769672)

In specific cases, the Block Else segment was not executed.

Where Clause Using Link Variables (QCR # 775298)

Changing Link Query to Link Join resulted in an incorrect Where clause if the Where clause was based on variables from the link.

Combo Box on .Net Browser Client - (QCR # 775922)

The combo box in a .NET browser client application did not handle special characters properly.

EOL Characters in an XML Document (QCR # 783950)

Unnecessary EOL characters were added to an XML file generated by the XMLWrite program of an XCG component.

Lock on DB2 (QCR # 925141)

In certain cases, the runtime tree was not aborted when it encountered a lock in the database.

XML Namespaces (QCR # 968208)

The Read_XML program of an XCG component read the XML data, although the XML URI did not match the target namespace of the XML schema.

Focus Lost from Child Tasks (QCR # 972296)

Focus on a child task was lost after switching between different applications.

“Order By” of an Oracle Select Statement (QCR # 985771)

The unique key segments were unnecessarily added to Oracle select statements.

PPD Procedure on Runtime Plugs (QCR # 985771)

The PPD procedure failed on runtime plugs.

XML Namespaces (QCR # 986918)

The namespace indication was not set properly in an XML document generated by an XCG component.

Clicking on Phantom Tasks (QCR # 988820)

Clicking on a phantom task always entered the first control of the first record, as opposed to the clicked record and clicked control.

MLS on Field’s Picture (QCR # 996326)

The translation file incorrectly affected field picture strings.

eDeveloper 9.4 Service Pack 3c

New Features and Behavior Changes

XML – Support CDATA tags

The XMLGet function now supports the CDATA content. To write a CDATA portion in an XML tag you should simply place the entire CDATA sequence as the tag's value.

Menu Icons – Modified Display

In previous releases, menu icons were not displayed properly if the icon colors template did not match the colors template used by eDeveloper's internal toolbar images.

EvalStr function – Long String Variables

In previous versions the permissible length of a function's string parameter was up to 256 characters. The permitted length has been increased to up to 32,768 characters.

XML Functions – Slash as a Delimiter

By default, the period character is used as a path delimiter. This version now also supports the '/' character as a path delimiter when a defined path or element start with this character.

Fixed Problems

Browser Client – Overriding a Frame (QCR 13813)

A crash occurred when overriding a frame containing a browser client task using a new URL. Now, when you switch to another URL, the browser task is closed before opening the next URL.

Browser Client – Calling Child Tasks in Frame (QCR 758309)

When you called a child task in a different frame, the focus remained on the child task instead of remaining on the parent task.

Java Support – Sending Large Byte Arrays (QCR 934562)

When you passed a large byte array to an EJB, a java.lang.OutOfMemoryError exception occurred.

Java Support – Integer.MIN_Value (QCR 910060)

The Java-related functions did not handle the Integer.MIN_VALUE integer value correctly.

Escaped Characters in J2EE (QCR 988105)

Special characters returned from an EJB were returned in their “escaped” sequence and not as the actual characters.

Direct SQL – Parent Variable in Where Clause (QCR 288813)

The checker issued a warning when you used a parent task’s variable in the Where clause of a Direct SQL task, even though this is in fact permitted.

Click on Tree Control (QCR 308993)

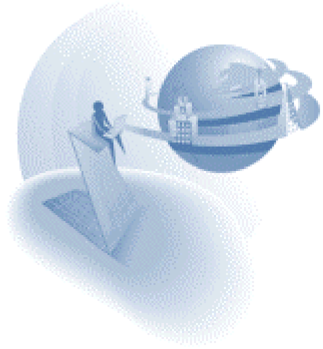
The Click operation did not work on a tree control when you clicked this control from a different task.

Phantom Tasks – Endless loop (QCR 989200)

In some cases when you clicked a phantom task that you defined to be closed immediately, the engine entered into an endless loop.

Phantom Tasks – Double Handler Execution (QCR 308981)

An event triggered by entering a phantom task window caused the corresponding handler to be executed twice.



eDeveloper 9.4 Service Pack 4

Behavior Changes

eDeveloper Error Log – Specifying File Name and Path

In previous versions, the error log file created by the engine to log all errors that occurred during application execution was created by default with the name **MGError.log**, within the eDeveloper engine's working directory.

In SP4 you can now control the error log file name and path using a new Magic.ini entry: **GeneralErrorLog**.

To define the error log file name and path, enter a path and file name in the entry, for example:

GeneralErrorLog=C:\MyLogs\MGError.log

If you leave this entry blank, an error log file will not be created.

Note: When you upgrade to SP4, your existing .ini file is not updated and therefore the **GeneralErrorLog** entry is not defined. This causes the engine not to create the error log file.

To define the engine to continue and create the error log, enter the following line in your Magic.ini file:

GeneralErrorLog=mgerror.log

Changing the Printer in the Main Printer Dialog (QCR #301328)

Changing the printer in the main printer dialog box opened from the **File** menu affected programs that print to the default printer.

The Printers dialog is only used to modify printer settings but not to change the default printer. Programs that print to the default printer will continue printing to the system default printer.

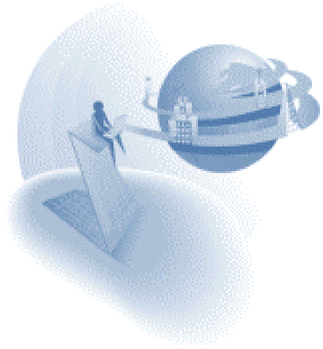
Clicking on a Non-parkable Element

In previous versions, clicking the form background or a non-parkable control made the engine exit and re-enter the current control and execute the entire control cycle (Change, Verification, Suffix, and Prefix).

In SP4 this behavior has been corrected and now the control only exits its Edit mode and only executes the Control Change handler.

Control Cycle Re-executed When Clicking Phantom Tasks (QCR #771650)

The Control cycle (Prefix, Verification, and Suffix) was re-executed when clicking a phantom task.



Hyphens in a Check Box Control (QCR #297963)

A Check Box control did not display its text if the text was set by an expression and the text string contained a hyphen.

Switching Between Phantom Tasks (QCR #297493)

Switching to a Phantom Task by clicking a record that was not the first record always set the focus in the Phantom Task to the first record.

Logon Function in Main Program (QCR #571895)

A user logged on by the Logon function in the Main Program was not retained as the logged-on user for the duration of the application.

Modal Batch Task Window Not Refreshing its Display (QCR #587893)

A Batch task window did not refresh its view when opened as a Modal window.

EJB - Interface change

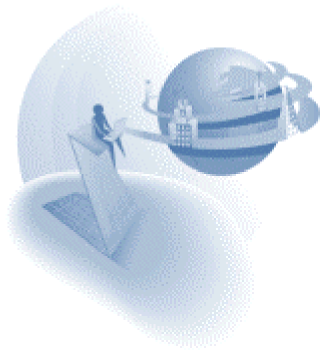
The eDeveloper EJB builder has had a face-lift and is now in the format of a wizard.

For more information, please refer to the XCG chapter.

Java Messaging (JMS) and Java Component Generator (JCG) – Modification

The Java modules that are used by the Java Component Generator and the Java Messaging component are now provided in the following file: **Support\edevUtils.jar**. If you upgrade to SP4 from a previous service pack you should manually modify the CLASSPATH entry under the [MAGIC_JAVA] section in the INI file to include the new path. For example:

```
[MAGIC_JAVA]
CLASSPATH=Support\edevUtils.jar;.;
```



New Features and Features Enhancements

View Refresh Event Enhancements

In previous versions, the View Refresh internal event always attempted to remain parked on the last parked record by re-locating the record. The View Refresh internal event has been enhanced to let you control the locate procedure by passing an argument in the Raise Event operation.

For more information on the View Refresh event, see the *View Refresh* Help page.

Flow Properties in Handlers

The Flow Mode and Flow Direction properties available for Record Main operations are now also available for operations in handlers.

These properties are available in these handlers:

- Control Prefix *(Only Flow direction)
- Control Suffix *(Only Flow direction)
- Control Verification
- User event
- Internal event
- System event

Revert – New Mode for Verify Operation

You can define a Verify operation to produce an Error-style message box, as with the regular Error mode, by using the Revert mode. However, unlike the Error mode, instead of exiting the handler where the Verify operation is executed, the Revert mode instructs the engine to re-execute in reverse order all operations that preceded the Verify operation in the handler.

The Revert mode is only available for these handlers:

- User events
- System events
- Internal events
- Timer
- Expression
- Control Prefix
- Control Suffix
- Control Verification

Note: The Revert mode behavior simulates the behavior of executing the Error mode in the Record Main.

Editing – New Force Exit Level

The Force Exit property of a User event supports a new level: Editing.

The Editing level instructs the engine to exit a control's Edit mode before executing the corresponding handler.

For more information on the Force Exit property of the user event, see the *eDeveloper Reference Guide.*

MainLevel – New Function

The new MainLevel function is similar to the existing Level function. However, the MainLevel function only returns a task's main levels, i.e. Task, Group, and Record.

For more information on this function, see the *eDeveloper Reference Guide.*

System Environment Variables – Retrieval and Setting

Two new functions: **OSEnvGet**, and **OSEnvSet** let you retrieve and set the values of system environment variables.

For more information about these new functions, see the *eDeveloper Reference Guide.*

Retrieval of a Directory's File List

A new function, **FileListGet**, lets you easily retrieve the list of files in a designated folder.

For more information about this new function, see the *eDeveloper Reference Guide.*

Identifying the First Record Cycle in a Task

The new **IsFirstRecordCycle** function lets you identify the first record cycle in the task. This information is useful for running logic that on the one hand is based on the already-fetched data, after the Task Prefix, and on the other hand for logic that is run only once, on entering the task.

For more information about this new function, see the *eDeveloper Reference Guide.*

Default Push Button

A new GUI Display form property lets you define a push button as the form's default push button. This lets you activate a default push button directly by pressing the ENTER key without the need to reach the push button explicitly.

For more information about this property, see the *eDeveloper Reference Guide.*

Transparent Background in Read-only Mode

If an Edit control is defined with a transparent background color, when you park on such a control in read-only mode the background color becomes opaque. This is to indicate that the control is the control in focus.

Prior to SP4, the opaque color was always the 3D Object system color and you could not modify this color. In SP4, when you set a background color as Transparent, you can now also select the background opaque color.

When you park on a read-only Edit control defined with a transparent background color, the opaque color you selected will be the background color of the parked control.

Note: Colors that were set as transparent in previous versions will retain the 3D Object system color as their Opaque background color.

Row Highlight

The GUI Display form's Table control lets you easily define how the current record in the table should be highlighted. You define the row highlighting by using the following two new Table control properties:

- ❑ Row Highlight Style
- ❑ Row Highlight Color

For more information about these new properties, see the *eDeveloper Reference Guide.*

Keep Tree View – New property

When you place a Tree control in a phantom task, on reentering the phantom task, the Tree control display is reset, even if the range of records is not modified.

In SP4 you can instruct the Tree to keep its view and state on reentry by using the Tree control's new **Keep Tree View** property.

For more information about this new property, see the *eDeveloper Reference Guide.*

Tree Control – Proper Handling of Drag-and-Drop Operations

SP4 supports the Drag-over and Drop phase of the Drag-and-Drop operation when performed on a Tree control.

The Tree control is now sensitive to the Drag-over phase so that the Tree control responds to the pointer in Drag mode as it is placed and moved over a Tree control. Moreover, on performing the Drop, the dropped-on node becomes the current node so that the Drop operation can be completed.

Packing and Unpacking Shared Values

Shared Values can now be passed from one eDeveloper process to another by packing and unpacking the Shared Values using these two new functions:

- ❑ SharedValPack
- ❑ SharedValUnPack

For more information about these new functions, see the *eDeveloper Reference Guide.*

Browser Client – Execution Failure Notification

In previous versions, when you ran a Browser Client application and for some reason the Browser Client modules failed to load the client, the user didn't receive any proper indication that the application was unable to run.

In SP4, a Browser Client page that fails to load the Browser Client modules invokes the **BCMissingModules** JavaScript function that can produce the proper indication.

You can place your own Java definition of this function on your Browser Client application pages to provide the feedback you require.

WAP Support – Default HTML

In previous versions, the default HTML page returned to indicate a request fault always returned with an HTML content indication in the HTTP header.

To enable other content indications, such as WML, the default HTML feature is now sensitive to the defined file name's file extension. If the extension is WML, the HTTP header will indicate the content as WML.

Broker Activity Log

In previous versions, the Broker activity log file created by the Broker was created by default with the name mrb_event.log.

In SP4 you may explicitly define the file name and path of the Broker activity log.

Use the new INI setting, ActivityLog, of the broker INI file to set the file name and path of the log file. For example:

```
ActivityLog = C:\MyLogs\broker.log
```

Tools Infrastructure – Object List Export

A new operation, **ObjectList**, has been added to the Tools infrastructure. This operation instructs the Toolkit engine to export the entire list of a given repository.

For more information about this new operation in particular and the Tools Infrastructure in general, see the *eDeveloper Reference Guide.*

Swedish Language Support

SP4 provides “const” files to support the Swedish language.

MailConnect – Support of Non-Default Port Number

The second parameter of the **MailConnect** function supports the **URL:Port** format. This lets you define a port number that is not the mail server’s default port.

For more information about this function, see the *eDeveloper Reference Guide.*

Mapping Magic BLOB RTF to SQLServer TEXT

You can now map an eDeveloper BLOB to an **SQLServer TEXT** data type. To map in this way, enter TEXT in the **Type** property of a **BLOB** variable. The recommended method of inputting data into a TEXT field is either by using the **Update** operation or the **File2Blb** function.

When using this data type in a Direct SQL statement, the maximum blob size will be 64K. In order to change that size (either increasing or decreasing it) you can use the **SQLBLOB = size** keyword in the database properties.

Note: When you retrieve data from this field, eDeveloper adds an extra byte to the end of the BLOB. When you use the **Blb2File** function to save the BLOB to a file, the file size will be one byte larger than the original size.

UDDI Interface Change

An interface change has been implemented in the search mechanism to improve performance when fetching multiple services from a UDDI server.

MSMQ Acknowledgement queue

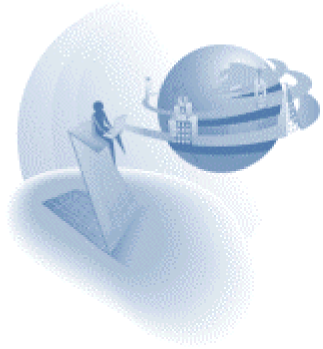
You can now send a queues message acknowledgement to a queue other than the MSMQ default. You can also now read the acknowledgment messages from this queue.

For more information see the *Messaging Connectivity* guide.

Application Installation Utility – User-defined Variables

A new section has been added to the Application Installation Utility (AIU) that lets you add your own variables to the AIU.

For more information see the *Application Installation Utility* guide.



Application Installation Utility – New Keywords

Some new keywords have been added that let you fetch information from the server on which you are installing an application.

For more information see the *Application Installation Utility* guide.

XCG Support for Paths Beginning With "/"

As the XML functions now support "/" when you define a path, XCG-generated components have been amended so that they use "/" instead of ".", which was previously used.

For more information about the XCG, see the *eDeveloper Reference Guide*.

J2EE - Access to EJB Invoker Credentials

eDeveloper J2EE support has been enhanced so that you can access the Principal information (credentials) of the user or application that invoked the eDeveloper-generated EJB method. To access the Principal information, you should define a BLOB variable for storing a Java reference and update the variable with the value of the expression: `GetParam(‘EJB_PRINCIPAL’).`

The EJB deployment descriptor includes a new environment entry: **PropagatePrincipal**, which controls the enabling of this feature.

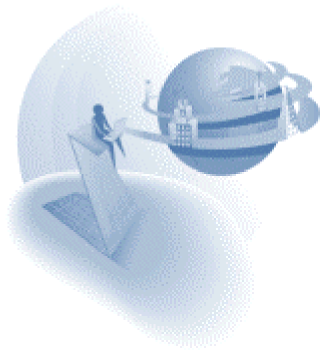
For more information about this enhancement, see the EJB-request Credentials section of the *eDeveloper Reference Guide*.

J2EE - Passing Java Objects

eDeveloper J2EE support has been enhanced so that EJBs generated by the eDeveloper EJB Builder support Java objects as arguments and return values. The EJB Builder's **Arguments** screen lets you specify Java classes for arguments and BLOB-type return values. For each argument/return value, if the **Java type** is set to **Object**, you can specify the complete Java class name, for example `java.util.Vector`. The Java class name specified should implement **the java.io.Serializable** class.

When the eDeveloper Enterprise Server receives a request from the EJB, each BLOB parameter corresponding to an argument mapped to a Java object, will automatically be set to a Java reference of the passed object. To return a Java object, the BLOB return value should be set to a Java reference to a Java object.

For more information about this enhancement, see the *eDeveloper Reference Guide*.



Fixed Problems

Browser Client & ODBC – Crash on View Refresh (QCR #4413)

A Browser Client application used to crash while performing View Refresh when working with a MySQL database through ODBC.

Browser Client – Updating Controls in Modal State (QCR #6144)

Radio buttons could be modified while a Modal window such as that of a Verify operation message box locked the page.

Working with an Oracle Unicode Database (QCR #7189)

When working with a UTF8 Oracle database, eDeveloper sometimes returned an Oracle error, "ORA-01461: can bind a LONG value only for insert into a LONG column". This happened when trying to update an Alpha field.

Numeric Arguments in Call Remote (QCR # 127713)

Numeric arguments did not return a correct result when passed in a Call Remote operation.

Missing Mail Body Text – (QCR #129647)

When retrieving a large number of email messages, the body text of some messages could not be retrieved.

Multiple Selection List Box – Clearing All Selections (QCR #139036)

A multiple selection List box variable was not updated when clearing all selections.

MouseOver & MouseOut in KBGET Function (QCR #166693)

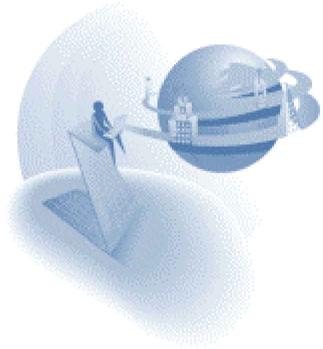
The Expression Editor rejected the use of MouseOver and MouseOut events in the KBGet Function.

Multi-marking of Tree Nodes Using the Mouse (QCR #276379)

Multi-marking of tree nodes could not be performed using the mouse.

Crash When Using Variable Functions in a Single Expression (QCR #280488)

eDeveloper crashed if the VarAttr, VarInp, and VarPic functions were used together in a single expression of a control’s picture.



Retrieving the Context Name (QCR #283670)

The CtxGetID function failed to retrieve the context name after it was set for the second time using a shorter name.

Checker Error in DB SQL Where (QCR #288813)

The checker gave an incorrect error when ranging on a parent task's Real variable in a DB SQL Where clause.

Cross Reference During Team Development (QCR #292891)

The Cross Reference procedure locked all other stations working on the same application in Team Development.

Split Window Subtask (QCR #298151)

A subtask, displaying a split window, was not properly refreshed when switching between subtasks.

Wrong Cross Reference Result (QCR #304307)

The Cross Reference utility failed to report a reference to a Variable used in an expression containing an application Right that the user did not possess.

Support for different Decimal Separators in XCG (QCR #309070)

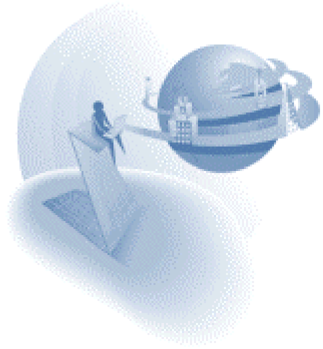
When the decimal separator of the eDeveloper environment was not a decimal point, this caused problems in the XCG-developed component when using numeric values that had decimal values. The Write program in the component now updates decimal separator with a decimal point in the task prefix and returns the user-defined value in Task Suffix.

XML Function on XML Direct Access (QCR #309724)

XML functions did not work properly for an XML Direct Access IO device if the XML file was provided through a URL.

Oracle Error When Having Both a Link Join and Link Query (QCR #406210)

When a program containing both a Link Join and a Link Query was executed, an Oracle error was generated "ORA 01480: Trailing Null missing from STR bind value".



Right-click on Multi-marked Tree Nodes (QCR #420901)

A right-click on multi-marked Tree nodes cancelled the multi-marking sequence.

Mixed Character Set in Log Files (QCR #431803)

In some languages, the engine log file was created with some of the strings written in OEM instead of ANSI.

Long Object Names in Selection Lists (QCR #443013)

Long object names, usually a result of an object name concatenated with a component name, were cut in selection lists. ToolTips now show the full name.

Model Referencing a Model (QCR #445115)

A model referencing another model had a mixed-up reference if more models were added.

Browser Client – Dynamic Combo Boxes on .NET (QCR #450887)

Combo boxes were not properly refreshed with new items when executing a Browser Client application using the .NET modules.

Crash When Using the XMLExist Function (QCR #448171)

The engine crashed in specific scenarios in which the XMLExist function was used.

Connecting to Multiple Databases in MSSQL (QCR #490042)

Direct SQL did not work properly when trying to connect to more than one database at the same time.

Dynamic Printer Settings (QCR #557963)

eDeveloper failed to respond to modifications of the Printers settings at runtime.

Browser Client – Updating Main Program Variables (QCR #596189)

A Browser Client application did not perform properly after updating a Main Program's variable that was also set with an initial value.

Fractions in Call Remote Operations (QCR #685470)

Fractions were not passed correctly in Call Remote operations.

MGStations – Incorrect Results (QCR #695342)

The MGStations utility returned incorrect results if the license file was in the same working directory as the utility.

Updating an Alpha field with the Contents of a Rich Edit Field (QCR #695432)

When an Alpha field was updated with the contents of a Rich Edit field, the contents of the previous record were displayed instead.

Create Table Syntax Problem in Informix (QCR #708363)

There was a problem with the CREATE TABLE syntax when a table was created using eDeveloper's Informix gateway.

Large Views in MSSQL (QCR #730249)

The Checker rejected large record length views.

Crash when Toggling from Toolkit to Runtime and Back (QCR #743877)

eDeveloper crashed in some cases when toggling from Toolkit to Runtime and then back to Toolkit when the **ISAMTransactions** setting was set to True.

Control in a Hidden Column Appeared on Fit-to-MDI (QCR #754472)

A control set in a hidden column appeared when opening the window to fit the MDI.

XMLGet Function – Retrieving CRLF (QCR #758956)

The XMLGet function failed to return a CRLF sequence in an XML element.

Web Services Builder and Special Characters, Such as a Tab Stop (QCR #760506)

When the XSD of a document-style operation contained special characters, such as a tab stop, the builder misinterpreted these characters. These characters are now ignored.

Printing Tables with Hidden Columns (QCR #760616)

Tables were not printed properly if a column that was not the last column was hidden.

PDF Output in Background (QCR #764013)

The size of images in background engine PDF output differed from PDF output in a foreground engine.

Browser Client – VecGet of Context Names (QCR #775922)

A Browser Client application using the .NET Modules failed to retrieve a context name from the GetAllCtxName function.

DB2/400 Get Definition Problem (QCR #874898)

When performing a get definition of a physical file, the unique Logical Files were not fetched.

MSSQL – Get Definition of a Table with a Descending Index (QCR #929771)

A descending index in an MSSQL table was created as an ascending index after getting its definition.

AS/400 Get Definition Problem (QCR #931350)

There was an incompatibility problem with previous versions when performing a Get Definition of a table. eDeveloper now fetches the "column name" of a Physical Table instead of the "column heading".

Browser Client – Task Prefix Execution (QCR #978290)

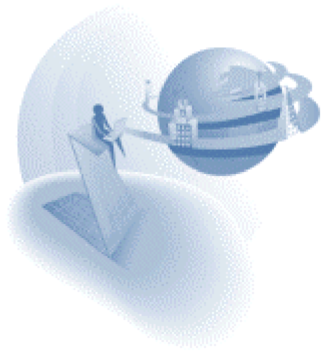
In some cases, a called Browser Client task’s Task Prefix handler failed to execute.

Upper Function in French Language (QCR #984141)

The Upper function did not work properly for the French language

eDeveloper inserted 32000 Characters to Oracle LONG Data Type (QCR #987931)

When an alpha field was mapped to LONG, eDeveloper always appended blanks, thereby inserting the full 32000 characters.



Misplacement of Floating Palettes (QCR #989509)

The Toolkit palettes were reduced to zero width and height after opening and closing the Export\Import dialog box.

Wrong Cross Reference Result (QCR #997028)

The Cross Reference utility failed to report a variable reference used as an argument in a Call Task operation.

Passing Numeric Values in Call Remote Operations (QCR #999602)

Numeric values were not passed properly from one background engine to another using the Call Remote operation.

eDeveloper 9.4 Service Pack 4a

Fixed Problem

Default Push Button – Verify Error

A Verify Error operation was not properly reflected when it was defined in a Control Verification handler and a default Push button was activated.

eDeveloper 9.4 Service Pack 4b

Fixed Problem

J2EE - access to EJB principal (QCR # 936306)

Accessing an EJB principal using GetParam('EJB_PRINCIPAL') did not return a valid Java reference object if another task was called before evaluating the GetParam function.

Loading the MGRQSOAP.DLL (QCR # 283786)

The SOAP gateway failed to load even if its path was defined in the system environment PATH setting.

Browser Client (QCR # 757131)

Using the .NET modules, an improper result was observed on calling two batch tasks that returned HTML pages.

INIPut function (QCR # 301477)

Modifications of the [MAGIC_SYSTEMS] section in the INI were not reflected in the Application list.

Oracle Connection – (QCR # 750689)

When no user name and password were defined for an Oracle database, the Login screen was repeatedly opened for every access to the database for browsing a table.

Browser Client – Combo box (QCR # 989800)

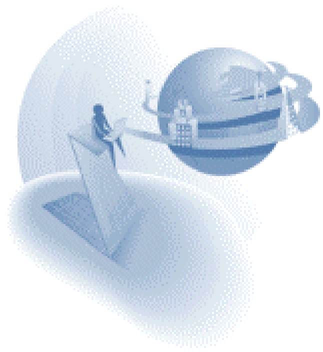
A combo box control that is ranged using a main program variable was cleared on switching to Create mode.

Broker – Loading Engines (QCR # 776751)

The broker could not load engines, if the broker was executed from a directory that was not the engine's directory.

Call COM Operation in Task Prefix (QCR #778945)

Call COM operation was not properly executed when it was defined in the Task Prefix and the Active-X control was defined with Persistent Data.



eDeveloper 9.4 Service Pack 4c

Fixed Problem and Changes of Behvaior

COM Interface – Blob Support

The COM Interface builder has been enhanced to let you create methods with Blob parameters.

Variant Functions– Blob Support

The **VariantGet** function was enhanced to properly extract arrays of UI1 as a Blob value.

MTblGet Function – Performance

The performance of the MTblGet function was improved in this release.

Modifiable Property Computation (QCR # 260741)

The Modifiable property of a control was unnecessarily recomputed.

INIPut of a Printer (QCR # 284709)

The [MAGIC_PRINTERS] section of the Magic.ini file was not properly modified when you used the INIPut function.

COM Support - Negative Values (QCR # 438258)

Negative decimal values were not returned properly from a COM object.

Web Service Support - Negative Values (QCR # 491655)

Negative decimal values were not returned properly from a Web Service call.

INIPut of an Application entry – (QCR # 617868)

The Application list did not reflect modifications of the [MAGIC_SYSTEMS] section that were performed using the INIPut function.

Browser Client – CallProgURL (QCR # 747767)

A browser window that was opened as a result of a CallProgURL function displayed the browser toolbar.

XML Functions in Background Mode (QCR # 773148)

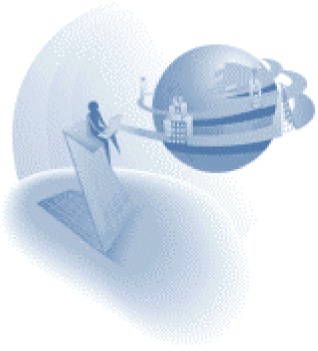
In some cases, when evaluating XML functions using a background engine a runtime exception error occurred.

Browser Client – Java-based Module (QCR # 776590)

In some cases the browser became stuck when running a Java-based browser client application.

Authenticated Web Service (QCR # 787686)

When accessing a Web Service that requires authentication, the service provider did not properly receive double byte characters.



eDeveloper 9.4 Service Pack 4d

Changes of Behvaior and New Features

Browser Client Modules Installation on Windows XP SP2

The installation utility for installing the Brower Client modules on the client side works properly also on Service Pack 2 of Windows XP.

Single Thread and Multi Thread OLE Automation Using Mg_ocx.dll

The Mg_ocx.dll's **ObjectLoad** function for direct handling of OLE automation has been modified to serve single thread usage.

If you want to perform OLE automation on a COM object in a multi-threaded constellation where the handle of the same object is used by more than one thread, you should use the new **ObjectLoadMT** function.

The **ObjectLoadMT** function syntax is the same as that of the **ObjectLoad** function.

Access Key of a Disabled Menu Entry

By definition, a disabled event cannot be activated by pressing the access key set for the event. This worked correctly when the event entry was explicitly disabled. However, this was not the case when a menu entry was implicitly disabled, for example by disabling the parent menu entry or by opening a modal window that disabled the entire pull-down menu.

The behavior is now consistent, so that whenever a menu entry is disabled it cannot be activated using its access key.

Browser Client - Clicking on a Non-parkable Element

In previous versions, clicking a non-parkable control made the engine exit and re-enter the current control and execute the entire control cycle (Change, Verification, Suffix, and Prefix).

In SP4d this behavior has been corrected and now the control only exits its Edit mode and only executes the Control Change handler.

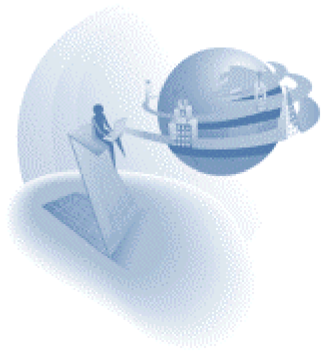
Table Control – Alternating Color

The Table control has been enhanced with two new properties, **Set table color by** and **Alternating BG color**, which let you set alternating background colors for table rows.

To set the alternating background color you have to first define a color for the table. You do this by setting the **Set table color by** property to **Table** and then define the **Alternating BG color** property.

Note: The Alternating BG color property is not yet supported for a table control set as Table in window**.**

For more information about these new properties, see the *eDeveloper Referenc*e.



Affecting the Background Color of 3D GUI Elements

You can use the **Special3Dstyle** special flag to define 3D GUI controls with the background color set by a control's Color property, as opposed to using the standard Button Face system color.

The flag should be placed in the Magic.ini file's [MAGIC_SPECIALS] section and you should set the flag to **Special3Dstyle=Y**. When you set this flag to **Y**, you can also define a color for other 3D GUI elements, such as the Menu bar. You should also be aware of the following implications:

- When you set the Text, Rectangle, Ellipse, Group, Check box, Radio button, and Tab controls to 3D style, they use the background color set by the Color property to paint the control's surface.
- Color entry #50 sets the background color of the column headers of a Windows 3D table, the surface of the splitter, and the background color of a disabled 3D Edit control.
- Color entry #51 sets the foreground and background colors of the Menu bar.
- Color entry #52 sets the foreground and background colors of the Toolbar.
- Color entry #53 sets the foreground and background colors of the Status bar.

Note: This flag affects both the runtime and toolkit screens.

XML Document Encoding

Two new functions, **XMLGetEncoding** and **XMLSetEncoding**, let you retrieve and define the encoding of an XML document.

For more information about these new functions, see the *eDeveloper Reference*.

Text on Image Button

A new style for the Button control lets you define an Image button and its accompanying text.

When you set the style of the Button control to **Text on Image**, the text for the button is defined according to the value of the Format property, and the button image is defined by the associated variable or the **Default Image file** property if no variable is associated with the control.

For more information about the new style, see the *eDeveloper Reference*.

EvalStrInfo Enhancements

The EvalStrInfo function supports additional modes of operation. The new modes let you retrieve additional information pertaining the given expression.

For more information about the function, see the *eDeveloper Reference*.

Fixed Problems

Inter-Frame Browser Client Call Operation (QCR # 293206)

A Call operation to a browser client task from one frame to another failed in some cases when the browser client application was executed using the .NET framework.

Decimal Separator Using Pervasive (QCR # 608742)

The Pervasive gateway failed to update a Numeric field if the decimal separator was not a period.

XML Document with a Non-English File Name (QCR # 778204)

eDeveloper failed to open an XML document file saved with a non-English file name.

Changing Range of a Resident Task with Link Join (QCR # 175729)

eDeveloper crashed when recalling a resident task, defined with a Link Join operation, when using new range settings on the Main Table.

Identity Field Using MSSQL (QCR # 186430)

When a table's Unique Key is an identity field and the table is set with a trigger that adds a record to another table by identity, eDeveloper displayed the identity of the second table instead of the Main Table.

Static-type Property of a Group Control (QCR # 988054)

The Static-type property of a Group control was disabled.

Modal Window Caption (QCR # 975343)

In the Form Editor, a modal window showed a regular window title size, whereas at runtime the window title size was smaller.

The Form Editor now shows the window title of a modal window using the same size used for it at runtime. Due to operating system limitations, this bug fix is only available for Windows XP using a non-Windows Classic theme.

Parked Control in a Table With Row Highlighting (QCR # 920718)

In a Table control defined with BG & Controls Row Highlight style, when parking on a modifiable control in Modify or Create mode, the control did not preserve its original color.

Split Window Icon (QCR # 760895)

The icon set for a window was not shown at runtime if the window was defined as a split window.

Overriding a Browser Client Page In a Frame (QCR # 756831)

The browser got stuck closing a browser client task displayed in a frame, when trying to externally override the frame content.

Browser Client Task from a Component (QCR # 744655)

The range of a browser client task was not properly evaluated if the task was part of a component and the Call operation directed the window to '*_top*'.

Fit-to-MDI Split Window (QCR # 731357)

A Fit-to-MDI window set with a splitter did not prevent the end user from moving the window from its fitted location.

Calling .NET Based Web Services (QCR # 694776)

eDeveloper crashed when accessing some .NET-based Web services.

Link Outer Join on DB2 (QCR # 286473)

In some cases eDeveloper crashed when executing a program that performed a link outer join on DB2 tables.

Fit-to-Size on Image Button (QCR # 135703)

The fit-to-size command did not work on an Image button.

Remark Line After Block Operation (QCR # 115329)

eDeveloper failed to park on a control if a remark was defined just after a Block operation.

SetParam with Different Attributes (QCR # 404533)

eDeveloper crashed when setting a global value by a string just after setting the same global value by a BLOB.

Multiple Link Outer Join on DB2 (QCR # 979292)

Tasks with multiple link outer join operations on DB2 failed to execute.

Format Property of a GUI Output Edit Control (QCR # 972590)

eDeveloper crashed after modifying a GUI Output Edit control's Format property and then closing the form.

Browser Client – Chunk Size for No Main Table Task (QCR # 759098)

The default chunk size of a task with no main table was 30 instead of 1.

Browser Client – Special Character in Multi-line (QCR # 759098)

eDeveloper crashed on having a special character in a multi-line edit control.

Browser Client – Activating a Button (QCR # 750483)

The Record Suffix was executed upon pressing a button that is set to raise a user event.

Browser Client – Calling Merge (QCR # 773052)

The Browser Client application failed to open a merge page from an Expression event handler when running the application using the .NET modules.

Cross Reference on a DB Table (QCR # 151804)

eDeveloper crashed in some cases of performing cross reference operation on a DB table that is referred to via an expression.

Expression in HTML Merge Form (QCR # 305844)

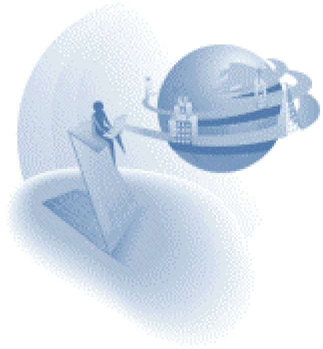
An expression used in the Picture property of an HTML Merge tag entry was regarded as unused and was reported as such by the Check Syntax utility.

Variable Name Length (QCR # 410182)

The length of a Select operation name property is now 31 characters long.

Scroll on Table with List Box Controls (QCR # 122398)

eDeveloper crashed on continuously scrolling up and down a table that has at least two list box controls.



EvalStr - Operands Limit (QCR # 304252)

The EvalStr failed in attempting to evaluate an expression string containing more than 99 operands. The limit is now 999 operands.

CtrlGoto & Check Box Controls (QCR # 751848)

Evaluating the CtrlGoto to set the focus on a check box control also toggled the check box control’s value.

Automatic Processing (QCR # 745142)

eDeveloper crashed in some cases of using the **OverwriteTarget** switch of the Import command.

View Refresh on Tree Controls

In tasks having a Tree Control, the View Refresh internal event now refreshes the tree content by maintaining the current state of every node in the tree.

View Refresh in Direct SQL Tasks

In previous versions, the View Refresh internal event had no effect on a Direct SQL task.

In SP5, the View Refresh event now affects a Direct SQL task like a regular main table-based task, with the exception that View Refresh in a Direct SQL task cannot remain parked on the last parked record after the view is refreshed. As a result, the locate mode parameter value of zero, which instructs the task to remain parked on the same record, is not supported. Passing the value of zero as an argument of the View Refresh event results in locating the first record in the view. This is also the default behavior of View Refresh in a Direct SQL task when no argument is passed.

COM Support – Handling ActiveX Events Synchronously

ActiveX events are now handled synchronously. This solves problems where event parameters passed by the ActiveX control were not available in the eDeveloper event handler and where here was the inability to intervene during the ActiveX procedure.

This solves reported QCRs: #446976, #308869.

Handling Errors of an Abort Engine Directive

In previous versions, if an error was encountered that aborts the task, the entire runtime task tree was closed and any handler of the error event would not be executed.

In SP5, when an error aborts the task, the runtime task tree is only closed to the point above the task that opened that transaction. Furthermore, handlers of the error event will be executed properly.

XCG Change of Behavior

When the XCG tried to load a schema in which there was a possibility that there may be recursion, it stopped after 2 levels. This caused problems for schemas in which there was no actual recursion. In order to give the developer more control, we added a new entry to the INTERFACE_BUILDERS section.

```
[INTERFACE_BUILDERS]

XCG_RECURSION_LIMIT=5
```

If this entry appears, the XCG will stop after the number of levels in this property. If the entry does not appear, the XCG will stop after 5 levels

New Features

New Push Button Property – Park On Click

A new push button control property lets you on the one hand set the push button control to be parkable, but on the other hand skips the flow-based logic that might be executed when switching to that control. This setting is useful for a Cancel button, for example if you define a parkable push button with **Park on click = No**, the Cancel event is raised when clicking the button, without any further execution of control level handlers.

For more information about the Park on Click property, see the *eDeveloper Reference Guide.*

Blob2Req – New Function

The Blob2Req function is complementary to the File2Req function. The Blob2Req function lets you directly pass the content of a BLOB value back to the requester by avoiding the need for creating a file for each BLOB before returning it to the requester.

For more information about the new Blob2Req function, see the *eDeveloper Reference Guide.*

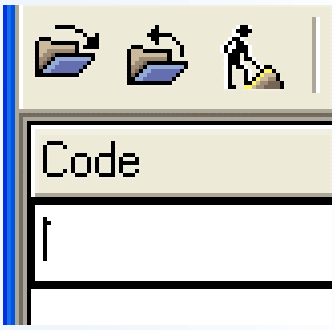
MDI Window Client Edge

By default, the MDI Window client edge, i.e. the inner border of the main window of the MDI, has a 3D style. In some form designs, such as those that are displayed as Fit-to-MDI and have 3D controls tightly located by the form's borders, the extra 3D effect displayed by the client edge produces an undesired look.

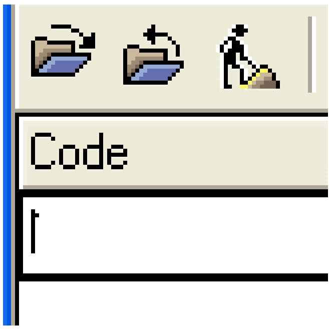
For these form types, you can now set eDeveloper to hide the client edge by setting the following entry in the Magic.ini's [MAGIC_ENV] section:

```
MDIClientEdge=N
```

The images below show the difference between the two modes:



The MDI client edge is displayed



The MDI client edge is not displayed

Splitter Style – New GUI Display form property

You can now define a GUI Display form's splitter to be displayed either as a 3D line splitter or a 2D line. For more information, see the *eDeveloper Reference Guide.*

Border Style of a Windows-3D Table Control

In previous versions, the Border Style property of a table control was only supported for a 2D style table control. The border style is now also available for a Windows-3D style table control.

Note: Currently the Thin and Thick border style options have the same effect for the Windows-3D table control.

Bottom Position Interval – New Table Control Property

The new Bottom Position Interval property lets you define how the table control should fit in a given size. For more information, see the *eDeveloper Reference Guide.*

Browser Client – Silent Installation

Whenever a browser accesses a browser client application for the first time, the browser client modules are automatically installed on the client side. During the installation, some installation-related windows appear. You can choose to hide these windows by adding a parameter to the URL entry of the browser client application.

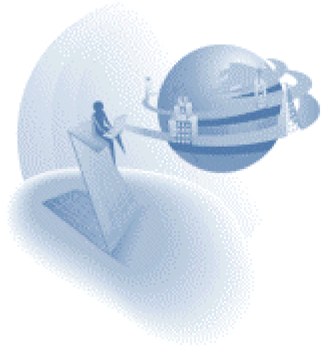
The parameter should be **Silent=Y** and you can add this anywhere within the URL. For example:

```
HTTP://MYWebServer/...?APPNAME=app1&PRGNAME=prg1&Silent=Y
```

Browser Client – Recovery of Delayed Requests

The eDeveloper Broker and Requesters have been enhanced to better handle slow network connections by automatically checking the state of a delayed request and operating accordingly. A new environment setting, **Browser Client Network Error Recovery Timeout**, sets the time interval after which the requester checks the state of a delayed request.

For more information on the new environment setting, see the *eDeveloper Reference Guide.*



Disabling the Check Index Mechanism in Deferred Transactions

The Check Index functionality helps you verify that there are no duplicate records in the database before exiting the record. This can be problematic when working with deferred transactions since the table might have been changed during the transaction without the changes being reflected in the database.

You can disable the Check Index functionality via a special flag as follows:

```
      [MAGIC_SPECIALS]
SpecialNoCheckIndexInDeferred=Y
```

Fixed Problems

Application Installation on Win98 SE (#771681)

The installation process failed if run from a local CD drive using AutoRun.

Sorted Column Indicator (#307022)

The sorted column arrow indicator remained visible after its column was hidden.

Manipulating the Last Row in a Table Control (#306704)

Manipulating the last row of a table control set to be a **Table in Window** caused the vertical scroll bar to appear when it should not have.

Range on Expression in ISAM Tables (#445387)

Ranging an ISAM table using an expression produced an empty record set, if the first fetched record did not match the range expression.

Automatically-Expanded GUI Output form (#751250)

A GUI Output form was expanded excessively if several expanded controls were defined in the table row.

Crash When Using Variable Related Functions (#998435)

eDeveloper crashed when evaluating Variable-related functions in a GUI Output-based task.

Browser Client Subform Using Local Sort (QCR # 692246)

In some cases a browser client task subform did not display its content when the sort was defined locally in the task.

Get Definition of a DB2/400 Table (#993869)

eDeveloper crashed when getting the definition of a DB2/400 table defined with various constraints.

Executing Oracle’s Stored Procedures (#816876)

eDeveloper crashed when executing stored procedures on an Oracle database.

Printing Thousands of Images (#301917)

When printing thousands of images using the Print Preview facility, eDeveloper crashed or produced an incorrect printing result.

Printing Page Number (#769172)

Printing the page number by updating a main program variable using the Page function failed to print the correct value.

Email Attachment File Name (#750062)

In some circumstances, eDeveloper failed to retrieve the file name of email message attachments.

Manipulating the Applications List (#301477)

The Application list was not properly refreshed to show the list of modifications that were introduced using the INIPut function.

Web Online with SUN JVM (#981492)

The hyperlink command did not work correctly on a Web Online page when it was executed on the SUN JVM.

Direct SQL Variables in the Flow Monitor (#731445)

The Flow Monitor displayed incorrect variable values defined for a Direct SQL task if these variables were not the first variables in the Record Main.

Phantom Public Tasks (#980238)

eDeveloper crashed when switching to a phantom task that was called from another application file using the Call Public operation.

Context Menu of a Rich Text Edit Control (#876109)

The context menu of a rich text edit control could not be opened using the mouse on a Windows 98 machine.

Large Numeric Values in Flow Monitor (#950727)

The Flow Monitor did not display large numeric values properly if their picture was defined with a separator for thousands.

Exiting a Child Task with Multi-marked Records (#746285)

Exiting a child task when multiple records were marked caused the exit event to be performed for every marked record.

Locate in Query (#282460)

Locate in Query activation did not work when both the Range and Locate criteria were defined on the same variable.

Web Service Builder –Invalid Elements Names (#772251)

The Web Service Interface builder did not indicate invalid element names, for example, element names that began with a digit.

Printing Images (#295547)

In graphical printing, images were printed smaller than expected.

Using the OverwriteTarget Switch of the Tools Infrastructure (#745142)

eDeveloper crashed when running the toolkit engine with an automatic processing sequence file that instructs the engine to import an object by overwriting an existing object.

XMLStr Function Performance (#282553)

The XMLStr function ran slowly on large XML portions.

Supervisor Group Name in Users File (#789503)

The Supervisor Group name used to be language-dependent. The group name is now language-independent.

HandledCtrl Function (#974699)

The name of a control handled by Control level handlers, such as Control Prefix, could not be retrieved using the HandledCtrl function.

Click on Combo Box (#302563)

In some cases the variable set for a combo box was not properly updated with the current combo box option when the combo box was manipulated by mouse clicks.

CtrlGoto Function in Phantom (#279678)

The execution of the CtrlGoto function in the Task Prefix of a phantom task took effect only when entering the task for the first time.

Mail Attachments (#445740)

In some circumstances, the content of an email attachment was corrupted.

Link Left Outer Join on DB2 Tables (#286473)

eDeveloper crashed when running a task that performs a link left outer join operation on DB2 Tables.

Browser Client – Replacing Tasks (#293206)

An error message appeared when switching from one task to another in a given destination.

Form Unit Manipulations (#745280)

Changing the form units of one form caused distortions in other forms of the same class.

Passing Values in Call Remote (#310582)

Values were not passed properly in a Call Remote operation if the two ends of the procedure were set with different settings for the decimal and thousands separators.

View Refresh After Incremental Locate (#126985)

eDeveloper crashed when performing a View Refresh after an incremental locate procedure. This crash occurred when the locate mode parameter of the View Refresh event was set to 1.

COM Support – Type Libraries List (#292319)

The Type library list viewed from an OLE variable did not include specific registered objects.

Authorized Menu Entries (#309098)

Menu entries in the pull-down menu did not refresh their authorization-dependent state after executing the Logon function.

Call Public to a Component Program (#991675)

A Call Public operation failed to invoke a public program from a component defined in the component repository.

Printing a Page Header (#749371)

Printing a page header form caused a change in the size of the Details form.

Right Click on a Combo Box (#420917)

Right-clicking a combo box control had the same effect as a left-click.

Printing Out to a Printer Driver of Sharp (#995312)

eDeveloper crashed when printing to a printer driver provided by Sharp Electronics.

Logical Field as an Index Segment on MSSQL (#432954)

eDeveloper did not permit a logical field to be set as an index segment for SQL Server 2000.

Uppercase in Dutch (#692741)

The Upper function did not work properly when working with Dutch.

Window Refresh of Batch Programs (#311399)

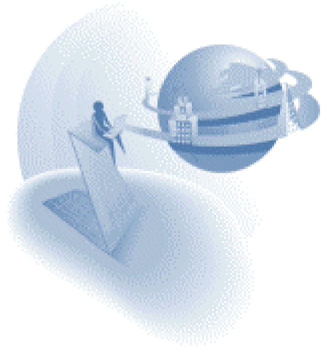
The window of batch programs set with Allow Events=No was not refreshed for the whole execution of the task.

Authorized Context Menu (#287469)

The context menu of a non-authorized user was opened if a program was called from a component.

Browser Client – Scrolling Through Records (#285889)

In some cases, verification handlers were not executed when scrolling through records.



Del Key on Multi-marked Records (#774372)

When you used the DELETE key, all records were unmarked even if the key was not assigned to the **Del Current Char** event.

Batch Delete on an ISAM Table (#261597)

A batch delete task did not delete table records when the returned value of the link was used to range the task view, and the locking strategy was set to immediate.

XCG Result Export File (#281236)

For some XML schemas the XML Component Generator created an incorrect export file.

Expanded GUI Output form (#982323)

eDeveloper crashed when printing a GUI Output form set to automatically expand according to its Multi-Line edit controls.

Error code of Duplicate Index (#312067)

eDeveloper showed the error code for a Duplicate Index error as #9 instead of #5.

Prompt Help (#992669)

The prompt help of a control was displayed on the status bar even after you left the control and parked on another.

Timer Event Handler (#298456)

A handler on a timer event was not executed if the timer value was set to 00:00:01 directly from the default value of 00:00:00.

Window Placement on Windows XP (#767235)

The placement of a window having a child window did not work properly on Windows XP.

Flow Monitor Messages (#339351)

The Flow Monitor displayed “Flow Call task: Calling" instead of “Flow Call task: Called" when a task was called.

Exit Operation (#442246)

The Exit operation failed to execute if the command was defined within quotation marks.

Scrollbars in RTL Browser Client Applications (#769876)

Redundant scrollbars were displayed in right-to-left browser client applications.

Browser Client – Paste From Context Menu (#3669)

Values were not properly updated in the field of a browser client application if the value was pasted by performing the paste operation from the context menu.

View Refresh on Direct SQL Task (#1812)

eDeveloper crashed when performing a View Refresh event in a Direct SQL task defined with an ORDER BY clause.

Cross Reference on a DB Table (#991134)

The Cross Reference utility did not indicate the use of a DB table by a program if the DB Table was both set as the main table of the task and linked to by a Link operation.

INIGet Function (#765522)

The INIGet function retrieved an incorrect value of an INI entry if a similar entry existed in a preceding section.

VarName Function in a Direct SQL Task (#769078)

The VarName function did not return a “Virtual” prefix.

Broker Startup (#297115)

The broker loaded too many instances of an application, if there were requests pending before the broker was loaded.

Browser Client – Modal Windows (#449002)

Parent task controls could be manipulated when the child task was opened as a Modal window.

Default Values in New Records (#975931)

In some circumstances, fields defined with default values were updated with incorrect values when creating new records.

Browser Client - Tabbing Out a Combo Box (#780858)

When tabbing out of an open combo box control, the focus was moved to the browser address bar.

Verify Error in a Component Program (#685996)

A Verify Error in a component batch program did not abort the task.

Wrong Returned Value of the JMS OpenDestination Program (#740193)

The OpenDestination program returned –9999, with no way of knowing that this was as a result of invalid authentication. The component now returns a Return Code of -1000 with the error message: "Open Destination failed due to authentication problem".

Bound Variables in Direct SQL Tasks (#771032)

eDeveloper crashed when running a Direct SQL task configured with bound real variables of the parent task.

Link Left Outer Join on an Oracle table (#747900)

The Link Left Outer Join operation was executed as a link inner join when locating on a non-indexed column when you set the “locate to” value.

XCG - User-defined XML Data Types (#245423)

The XML Component Generator indicated an error on a missing user-defined data type definition even though the user-defined data type was defined properly.

XCG - Handling Duplicate Elements Names (#663743)

The XML Component Generator did not properly handle XML schemas where different elements had the same name.

COM Interface Builder – Missing COM Creator (#304825)

The COM Interface Builder displayed an incorrect error message when the COM Creator object required by the Builder was not installed properly.

Printing Color Images to a PDF Driver (#312654)

Color images were printed in black and white when printed via a PDF driver.

Browser Client - Hourglass Cursor on Windows 98 (#306514)

The mouse pointer was not displayed as an “hourglass” when calling a batch task from a browser task running on a Windows 98 client using the Java-based modules.

Browser Client - Initial Value of the Default Class Property (#306729)

The initial value of the Default Class property of a control was evaluated only after the page was loaded.

eDeveloper 9.4 Service Pack 5a

Fixed Problems

Zoom Event of a Tab Control (#698)

The internal event of Zoom was not triggered on switching between the tabs of a tab control.

.NET bases Browser Client Application (#3349)

In some cases the client did not handle requests synchronously.

Recalling a Resident Task With Varying Range Settings (#175729)

eDeveloper crashed on recalling a resident task set with Link Join operations, and in each call the range would vary.

Loading WSDL information (#292854)

Some WSDL file could not be loaded properly from a Call Web Service operation.

Logon and Pulldown Menu (#294409)

The Pulldown menu set with various rights did not reflect the logged on user properly when the user logged on using the Logon function.

Import with Component's Rights (#308067)

eDeveloper crashed on importing an export file of an application that uses a component's right entry but no component entry exists in that application.

"One Page" Expand Form (#429072)

The size of an expanded form got increased more than required.

The XMLGetEncoding Function (#435593)

eDeveloper crashed on evaluating the XMLGetEncoding function set with an invalid parameter.

Undo Editing of a Child Form (#745415)

eDeveloper crashed on performing Undo Editing operation inside the form editor of a child form.

Browser Client - Blank Combo Option (#756679)

A combo box control on a browser client task did not keep a selected blank option.

Keep Table Width (#756679)

Not all columns of a table control were displayed even when the Keep Width property of a GUI table control was set to Yes, and the visibility of the columns was determined by variables updated in the task prefix.

Alignment of Numeric Fields in RTL Applications (#759731)

The default alignment of numeric fields in right-to-left application was set to Left.

Modifying the Terminal Environment Setting (#772626)

The Term() function that returns the Terminal environment setting did not reflect modification of the Terminal value using the INIPut function.

XCG - XML Schema Set to Include Additional Schemas (#772665)

The XML Component Generator did not properly parse XML schemas that were set to include additional schemas.

Using a Memory Table in a Resident Task (#980239)

eDeveloper produced wrong results when referring to a memory table in a resident task.

Browser Client – Large Tasks (#983063)

It took a long time to load pages of large tasks.

eDeveloper 9.4 Service Pack 5b

Fixed Problems

iSeries – Calling a CL\RPG Program (#986252)

A wrong error message was given on calling a CL\RPG program that terminated abnormally.

DB2\400– Scrolling After Locate (#303803)

Scrolling through the records could not be performed after locating on the first or last record on a DB2\400 table.

eDeveloper 9.4 Service Pack 5c

New Features

MSI Installation for .NET Browser Client Modules

The eDeveloper installation provides you with an MSI file that can be used on the client's side to install the Browser Client module for .NET. This module will show up as an entry in the ''Add/Remove Programs' list (accessible via the Control Panel).

Tool Windows Style of Modal Windows

In previous versions, modal windows were defined as "Tool" windows. This resulted in a certain uncomfortable behavior by the operating system: Alt-Tab would not switch to the next application, and eDeveloper's icon would disappear from the application icon list.

Moreover the caption for a Tool window is smaller than a regular caption. eDeveloper provides a special flag in order to remove the tool window style from modal windows. The special flag is:

[MAGIC_SPECIALS]

SpecialModalToolWindow = Y/N (Default = Y)

The result of setting this special flag to N is:

- The caption of all modal windows will be of a normal size, thus the remaining client area in these windows will turn out to be smaller in height.
- The icon for all floating windows will disappear.
- The uncomfortable operating system behavior regarding Alt-Tab will disappear.

Fixed Problems

Unused Namespaces in an XML Schema (#1293)

The XML component generator used to generate a program with syntax errors if the schema had unused namespace declarations.

Alt-Tabbing With eDeveloper (#284770)

When pressing Alt-Tab while having a dialog open, the control would stay in eDeveloper instead of jumping to the next application.

IO Timing (#284854)

eDeveloper crashed when setting the IO timing to 'On Demand' and using I/O by name.

RunMode() Return Value (#294422)

RunMode function returned 0 when called consecutively through a Main Program of a single request.

Error Message In Error Log (#299936)

No error message was found in the error log when eDeveloper failed to access the application.

2D Multi-line Edit Control Flickering (#300315)

When clicking on Multi-line edit fields with 2d presentation and color identical to the background color but different than white, there were occasional flickers of a white background.

Processing Error (#300579)

An 'In the parsing process' error message showed up after clicking on a certain URL, using the .NET Browser Client module.

Wrong Where Clause (#307314)

When linking to a table with a Unicode column, eDeveloper sent a blank padded value within the Where clause to the database, during consecutive links.

Group Elements in a Schema (#426779)

The XML component generator failed to parse a schema with group elements inside.

eDeveloper 9.4 Service Pack 5d

Fixed Problems

Link Query on Pervasive 8 via ODBC (#284729)

No record was fetched from a link query set for a Pervasive 8 table via ODBC when a record was located using an Alpha value.

Error Reading the MFF File (#299936)

The error log did not include any indication on failing to read the MFF file.

Wrong WSDL Schema Error (#688328)

eDeveloper reported an error "WSDL Schema Error" when trying to access a certain WSDL file.

Phantom Task Window Remains Opened (#740436)

A phantom task window that is opened from a batch task remains opened even after the parent batch task is terminated.

Text Based Printing (#740547)

eDeveloper used ANSI characters when printing to a text printer.

Browser Client – Check Boxes Values (#745317)

The value of a check box control got checked in a browser client program if the user clicked on the next line checkbox control and an error message was displayed.

Oracle Tables – Error on Insert (#747648)

An ORA-01400 was returned on attempting to insert a new record in a table set with a large alpha column.

Modal Window Border (#751954)

eDeveloper used to draw the border for a modal window with no title bar and no border.

Expression Handlers (#771212)

eDeveloper crashed on calling another program from a handler set on an expression based user event.

MSMQ – Memory Consumption (#774051)

Memory resources where not freed after performing open and close procedures of MSMQ,

Web service– Commented Operations (#920660)

eDeveloper crashed on using the WSDL Assist on an operation set with a comment.

Call Remote & Active Directory (#966880)

A server engine invoked by Call Remote, validated a passed user ID against the users file instead of the active directory it was set to work with.

Recompute of Link Join Condition (#997529)

The eDeveloper engine crashed when trying to locate within a task having a Link Join with link condition evaluated in Task level after refreshing the view and changing the link condition.

COM Support – Object Reference (#994995)

Object, which its reference was retrieved via a Call COM operation, was not properly released.

eDeveloper 9.4 Service Pack 6

Fixed Problems

Disabling Requesters' Audio Indication

By Default, the engine produces an audio indication for every request it handles.

You can disable this indication by using a new special flag under the [MAGIC_SPECIALS] section in the INI file, as follows:

```
[MAGIC_SPECIALS]
SpecialBeepOnRequest=N
```

Performance of the XMLGET function (#8281)

The performance of the XMLGet function in handling large XML data was significantly improved.

ODBC – Locating a Record that Includes BLOB columns (#209077)

eDeveloper crashed on locating a record in a table set with a column of a BLOB attribute.

DB2 – Accessing an Unavailable Database (#278788)

eDeveloper crashed on attempting to reconnect to an unavailable DB2 Database.

DB2 – SQL Range on Date Column (#279232)

eDeveloper failed to fetch records from a DB2 table on setting an SQL range on a date column

iSeries – Host IP Setting in Command Line Requester (#280478)

A command line request could not be submitted from an iSeries machine unless it was explicitly defined in the iSeries host table. Specifying the "Domain name server" address in the CHGTCPDMN iSeries command solves this problem.

Force Exit Event Property – Help Page (#282038)

The sequence of handling the Record level of the Force Exit property was incorrectly described in the properties help page.

Verify Revert Operation In a Block (#283155)

Placing a Verify Revert operation in a block operation caused an improper execution of the block.

Default HTML Page – Missing Arguments Values (#284238)

The Default HTML page that is displayed when no output is returned by the engine failed to display the values of the passed arguments if the arguments where passed via Submit.

File Input Error (#286835)

The flow monitor reported an error of failing to read the file when the input program contained a link operation.

Incorrect Tooltip Display (#287549)

In specific cases the tooltip set for a control did not display the correct string.

XML Component Generator – Handling Group Elements (#287667)

The XML Component Generator failed to parse schema files that include group elements.

Direct SQL – Rejected Valid Select Statement (#287967)

Valid Select statements having the Order By segment referring to the columns by ordinals were rejected.

Secret Name as Database Name (#290862)

Secret name could not be used as the Database Name of a database definition.

Default Button on a Subtask Form (#291586)

eDeveloper crashed on activating a default button set on a form of a subtask.

Oracle – Passing Long Strings to Stored Procedures (#292071)

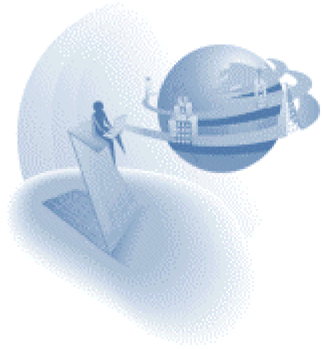
eDeveloper crashed on passing a parameter that is longer than 3993 characters to a stored procedure in Oracle updating a varchar2(4000) column.

View Refresh in Create Mode (#292218)

View Refresh event did not affect a task that was initially in Create mode.

Browser Client – Modifying Table's Records (#293361)

A Java script error was produced on modifying a record of a table that was already opened in an ancestor task in Read mode.



ActiveX Support – Missing control in ActiveX selection list (#293485)

A reported case of using the tssTaskPane ActiveX control showed that this control was not listed as expected.

Table Control - Table in Window – Reaching Table Columns (#293523)

In some cases, clicking on a column in a table control set to be “Table in Window” did not set the insertion point on that column.

Table Control - Alternate Color – Reflecting Foreground color of Controls (#299207)

A table control that was set with alternating colors did not reflect the dynamically modified foreground color of controls placed on the table.

XML – Modifying a <!CDATA> Element (#300382)

The XMLModify function done on an <!CDATA> element resulted in an insertion of such an element and not in a modification of the referred element.

Web Service – Parsing Large WSDL Files (#303076)

The WSDL Assist of the Call Web Service operation did not handle large WSDL files properly.

INIGET Not Reflecting INIPUT modifications (#303659)

The value returned by the INIGET function on an entry in the [MAGIC_PRINTERS] section does not match the value previously set by the INIPUT function on the same entry.

Opening a Task with Dynamically set Forms (#304894)

eDeveloper crashed on executing the a task with dynamically set form for display.

SplitterOffset Function’s Returned Value (#305148)

The SpliiterOffset function set to return the current offset of the splitter in percents did not return the correct values.

COM Support - Retrieving Arrays (#306411)

Arrays could not be retrieved from a COM object using the VecGet and VariantGet functions.

Table Control – Extended by Placement (#306610)

eDeveloper crashed on clicking on an unpopulated table row in a table control that was extended by placement factors.

Child forms – Modified by Placement (#307754)

Child form did not reflect placement modification according to the modification on the parent's window dimensions.

Oracle - Binding Values on Direct SQL Statement (#308176)

eDeveloper crashed on issuing an Oracle Direct SQL Statement if a string parameter containing the value to be sent in the LIKE statement was bound.

German Language – Letter Case Support (#308267)

Some characters where not represented properly in upper case.

Splash Screen With Logon Dialog (#310579)

The Splash screen was kept displayed as long as the Logon dialog was kept opened.

Component Programs - Initial Mode Set to As Parent (#421035)

Calling a component program set with an initial mode of As Parent gets the mode of the component's main program and that that of the calling program.

Hebrew – Utilizing the Special3DStyle Special Flag (#422258)

The status bar showed unexpected text in the Hebrew version when the Special3DStyle special flag is activated.

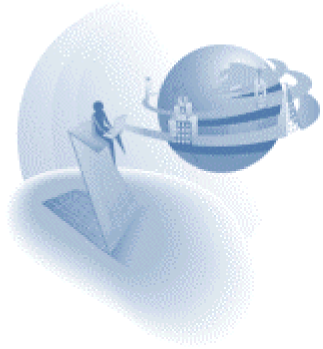
Inherit All Action (#423610)

Performing the action of the Inherit all on a control inherited properties that are not defined by the associated model.

Now, on performing the Inherit All action, the control will re-inherit only such properties that are defined by the associated model.

Browser Client – Row Height of a Table (#437807)

The row height of a line in a table got increased with every addition of a new record.



RQHTTPHeader function - Setting Cookies (#682666)

Only the last evaluation of the RQHTTPHeader function to set a cookie value took effect.

Term Function (#689436)

The Term function that is supposed to return the Terminal environment setting value did not return the correct value after the setting was modified using the INIPut function.

COM Support – Handling The Microsoft Forms 2.0 Combo Box Control (#695741)

Activating the AddItem method of the Microsoft Forms 2.0 Combo Box controldid not take effect on the control.

Application Installation Utility – Welcome Screen Display (#696502)

In the welcome screen, which is displayed by the installation procedure, did not replace the %PRODUCT NAME% tag properly.

Apache Requester – Reconnection to the Broker (#698131)

The Apache requester failed to reconnect to a restarted Broker module.

Format Property - Keeping Broken Values (#738298)

The Format property got reset whenever the data of the control was modified.

Call UDP – Passing Unicode Values (#739469)

eDeveloper crashed on passing a Blob value containing Unicode characters through a Call UDP operation.

Syntax Check on ExpCalc Function (#740360)

The syntax checker reported an error on any use of the ExpCalc function.

Browser Client - Replacing Tasks (#740579)

A program that is set to be opened in Create mode is opened in modify mode if it is set to replace an existing task.

XML Component Generator – Length Limitations (#741153)

For some schemas the XML Component Generator produced an invalid component due to internal length limitations.

Table Control – Border Display (#741966)

In some cases the border of the table control was displayed as an incomplete line.

Calling a Program During Delay Period (#743840)

A program called during a delay period could not be closed using the Escape key.

Printing – Multi Line Edit Controls (#745937)

A report was not printed properly if the reports form contained multi line edit controls.

Creating Flat Files (#746061)

eDeveloper created an invalid flat file (MFF) when the version control mode was activated.

UDDI Search (#746228)

Querying the same web service several times consecutively kept appending the service information.

Users File – Modifying User's details (#746966)

Once a user modified the user details, that user could not modify the user details again unless the user logged on again.

Alternate Color – Wrong Display (#749202)

In some cases the alternating colors where not displayed properly.

VariantCreate Function – Support ByRef Attribute (#749289)

The VariantCreate function did not support the extension of the variant data type to indicate that the value is set by reference.

Printing – Large Reports (#751265)

In lengthy reports the upper right-hand-corner display was partially displayed.

Repeated Re-computation (#752576)

eDeveloper crashed upon long consecutive re-compute cycles.

PervasiveSQL – Persistent User Details (#752597)

eDeveloper did not keep the username and password that were required for connection to a remote database, thereof the Logon dialog was displayed every time the database was accessed.

EditGet Function on Negative Numeric Values (#753444)

The Editget function did not return the correct value of a field, if that field contained a negative numeric value.

Browser Client – IFRAME Relocated (#755110)

The location of an IFRAME element was modified on opening a program in it.

Browser Client – Form Defined Inside a Table (#755900)

Defining a browser client form in a cell element set with center alignment caused the table row highlighting to keep its position relative to the browser window instead of the form.

Tools Infrastructure – Space Key Combinations (#756831)

The Space and Ctrl+Space key combinations did not work using the Key option of the simulate operation.

Oracle – Reporting Locked Record in Log File (#757115)

The mgerror log file did not include any indication of locked records when the engine was executed in background mode.

Text-Based Output - Line Control Display (#757127)

A vertical line control was not shown in the GUI output result if the line length was increased from its top coordinates.

Deploying Flat Files on Windows 2003 (#763865)

In some cases a flat file could not be used in deployment over a Windows 2003 operation system.

GUI Output – Printing Rich Text (#764727)

A Rich Text Edit control kept a size limit that was that of the initial value set of that control.

*GIF Support - *.GIF Filter Missing (#767726)*

The *.GIF filter of the Open File dialog used to select Images in Toolkit did not include the *.GIF filter to select GIF images.

DB2/400 – Not All Record Displayed (#771863)

Browsing a DB2/400 table in line mode set with a caching strategy did not show all available records.

Table Control – Image Button on Table In Window (#772659)

An Image style button placed on a Table control that is set to be a Table in Window, was not properly displayed.

Oracle – Changing The Connected User (#897332)

eDeveloper crashed upon reconnecting to an Oracle database as a different user

Link Condition on Group Level (#966283)

An operation was not executed properly if the link's condition was modified at the Group Suffix handler.

Image Cache Support (#976458)

Images were not cached according to the Image cache setting when an application is executed under the toolkit engine.

Error Handling in Resident Tasks (#981585)

Handling database related errors such as Locked Record were not properly handled in a resident task.

MSSQL – Link With No Index (#983306)

eDeveloper crashed on linking an MSSQL table by the lower representation of an existing value when the link operation is set with no index.

Right-to-Left Applications – Table Control Display (#987393)

A scroll bar appeared on "hiding" the right-most column of a table control in a right-to-left application.

Oracle – Handling Date Values in French (#988994)

eDeveloper generated a wrong SQL Where clause over a date value stored in an alpha field in an Oracle database , when using a French NLS_LANG.

Printing Images (#989236)

Printing a GUI Output form with a BMP file took an extensive time to be produced.

Nested Programs Execution (#989387)

eDeveloper crashed after extensive nesting of program called from an event handler.

eDeveloper 9.4 Service Pack 6a

Fixed Problems

SQLServer - Support Application Roles

An MSSQL database entry may now include a new instruction by which the connection will be done under a specified application role.

The role name and password should be defined in the database information as follows:

```
SQL_APPLICATION_ROLE = [Role Name]
SQL_APPLICATION_ROLE_PASSWORD = [Password]
```

You may use secret names to define the role name and password.

Browser Client – Component's Events (QCR #293544)

Component events where not propagated up to the host application.

Idle Function (QCR#740761)

The Idle function always returned zero.

Java Classes – Double Data Type (QCR #743833)

Whole numbers were not passed correctly if they were passed to a double data type of a Java class.

Oracle – Setting NLS_Trans Value (QCR #757926)

eDeveloper crashed on performing APG on a table set with NLS_LANG=AMERICAN_AMERICA.UTF8 and that includes CLOB column.

ActiveX – Click Events (QCR #770004)

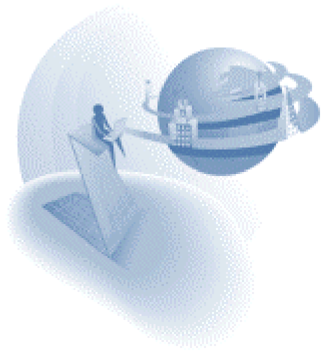
ActiveX controls did not respond properly to click events.

Web Service – Adding an Attachment (QCR #772569)

eDeveloper crashed on responding to a web service call by adding an attachment using the WSAttachmentAdd function.

Browser Client – Opening Tasks in IFRAMES (QCR #990308)

On calling the same task into the same IFRAME the Task Prefix handler would be executed on every second call.



Phantom Tasks – Closing Task After Sorting by Column (QCR #995310)

eDeveloper crashed on closing an orphan task just after the task data view was resorted by a sorting column.