

Miramo[®]

Automated Publishing

Release Notes

VERSION 9.2

Copyright © 2000 - 2012 Datazone Ltd. All rights reserved. Miramo® and mmChart are trademarks of Datazone Ltd. All other trademarks are the property of their respective owners.

Readers of this documentation should note that its contents are intended for guidance only, and do not constitute formal offers or undertakings.

'License Agreement'

This software, called Miramo, is licensed for use by the user subject to the terms of a License Agreement between the user and Datazone Ltd. Use of this software outside the terms of this license agreement is strictly prohibited. Unless agreed otherwise, this License Agreement grants a non-exclusive, non-transferable license to use the software programs and related documentation in this package (collectively referred to as Miramo) on licensed computers only. Any attempted sublicense, assignment, rental, sale or other transfer of the software or the rights or obligations of the License Agreement without prior written consent of Datazone shall be void. In the case of a Miramo Development License, it shall be used to develop applications only and no attempt shall be made to remove the associated watermark included in output documents by any automated method.

The documentation accompanying this software must not be copied or re-distributed to any third-party in either printed, photocopied, scanned or electronic form.

The software and documentation are copyrighted. Unless otherwise agreed in writing, copies of the software may be made only for backup and archival purposes. Unauthorized copying, reverse engineering, decompiling, disassembling, and creating derivative works based on the software are prohibited. This notice is provided for information only, and does not constitute a License Agreement.

Datazone does not warrant that the software will be free from error or will meet your specific requirements. You assume complete responsibility for decisions made or actions taken based on information obtained using the software. Any statements made concerning the utility of the software are not to be construed as unexpressed or implied warranties.

Trademarks

FrameMaker, FrameMaker+SGML, Acrobat, Adobe, PostScript, Adobe Illustrator and TIFF are trademarks of Adobe Systems Inc. Macintosh is a trademark of Apple Computer Inc. Windows NT is a trademark of Microsoft Corporation.

PANTONE is a registered trademark of Pantone Incorporated.

Miramo is a Registered Trademark of Datazone Ltd (see above).

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

Datazone Limited
Palm Gate, Greenane,
Killarney, Co. Kerry, Ireland.
Tel: +353 64 66 289 64
Fax: +353 64 66 289 65
Email: miramo@datazone.com
www.miramo.com

Release Notes (v 9.2)

Contents

Important Note	RNiii
System Requirements	RNiv
Solaris	RNiv
Windows NT	RNiv
C++, Java and .NET job processing APIs	RNv
Summary of changes in Miramo 9	RNv
mmServer Interface	RNvi
Processing Channels and Processing Channel Groups	RNvii
New licensing scheme	RNviii
Upgrades	RNix
New features (v. 9.2)	RNix
New features (v. 9.1)	RNx
New features (v. 9.0)	RNxi
APPENDIX—Processing channel throughput	RNxiii
DESKTOP relative performance	RNxiii
SERVER relative performance	RNxiii

Release Notes (v 9.0—9.2)

The capabilities in Miramo version 9.0 are a superset of the capabilities in preceding releases.

Many new features have been added to Miramo version 9.0 and some command line and code options have been changed from their form in earlier Miramo versions. Nevertheless nearly complete backwards compatibility has been maintained on a hidden basis, back to Miramo version 7.0.

While the functional architecture of Miramo 9.0 & 9.1 are backwards compatible with pre-Miramo 6 releases, the inline markup codes were changed significantly in Miramo 6.0. Upgrading a pre-Miramo 6.0 application to a Miramo 9.1 application requires markup conversion.

IMPORTANT NOTE

Though Miramo 9.0 & 9.1 work with FrameMaker 7.2¹ and above, the operating mode of Miramo 9.0 & 9.1 is significantly different in the case it is used with FrameMaker 8 or above. These Release Notes and the Miramo 9.1 documentation apply only in the case that Miramo 9.1 is used with FrameMaker version 8 or above. If Miramo 9.1 is used with earlier versions of FrameMaker the Release Notes and documentation for Miramo 8.1 apply (see [relnotes810.pdf](#)).

If Miramo 9.1 is used with versions of FrameMaker prior to FrameMaker version 8, these Miramo 9.1 Release Notes and the Miramo 9.0 documentation are invalid. If Miramo 9.1 is used with versions of FrameMaker prior to FrameMaker version 8 the Release Notes and documentation for Miramo 8.1 must be used. These are included in the folders: %MM_HOME%\docs\mm810 (Windows) \$MM_HOME/docs/mm810 (Solaris) The new features described in the Miramo versions 7 through 8.1 Release Notes also valid for Miramo 9.1 (see [relnotes810.pdf](#)).

1. Support for some features in Miramo 9.0 & 9.1 are missing when running with versions of FrameMaker earlier than FrameMaker 10.

SYSTEM REQUIREMENTS

The following assumes that Miramo 9.1 is used with FrameMaker version 8 or higher (see IMPORTANT NOTE on page [RN-iii](#)).

The primary factors affecting performance is the integer processing power of the host CPU and disk I/O. If very large documents are to be produced, in the multi-hundred or multi-thousand page range, then a large amount of RAM should be installed on the host. The exact amount depends on the complexity of the pages, and on the number of concurrent Miramo processes. SCSI and SAS disk interfaces are preferable to IDE and SATA interfaces.

Solaris

Miramo 9.1 runs on Sun Solaris SPARC systems (SunOS 5.8 and above) with FrameMaker version 8.

The FrameMaker Solaris license type must be 'Shared' or 'FrameMaker Server'. Running Miramo with a Unix 'Personal' license is not supported.

If directly-printed output is required, i.e. not via PDF, a PostScript[®] printer must be installed. The Miramo service and some features are not available on Solaris.

Windows NT

Miramo 9.1 runs on Windows on most versions of Windows NT 5 and NT 6, as follows:

- Windows XP Professional, Service Pack 3
- Windows 7
- Windows Server 2003, Service Pack 2
- Windows Server 2008 R2¹

x64 versions of the above operating systems are supported only when using FrameMaker 9 or above. All components of FrameMaker must be installed.²

A Windows NT 5 or 6 host should have at least 1 GB of RAM installed.

1. Running Miramo on pre-R2 versions of Windows Server 2008 is not supported.

2. The Acrobat Distiller or the Adobe PDF printer FrameMaker components must be installed before Miramo can be installed.

C++, Java and .NET job processing APIs

The C++, Java and .NET job processing APIs are available on supported versions of Windows NT.

- **C++**

Application interfaces must be built as a 32-bit applications.
For Visual Studio, the Platform must be set to Win32.

- **Java**

Java Development Kit (JDK) 1.7 or higher.
Application interfaces must be built as 32-bit applications and run using a 32-bit Java Virtual Machine (JVM).

- **.NET**

.NET version 4.0 or higher.
Application interfaces must be built as 32-bit applications.
For Visual Studio, the Platform should be set to Win32.

SUMMARY OF CHANGES IN MIRAMO 9

The following is a summary of the changes in Miramo 9.

- The Miramo Enterprise Module (**MEM**), no longer exists as a separate entity in Miramo 9.
- Previous **MEM** functionality is incorporated into *all* license versions of Miramo 9 as standard, at no extra cost.
- A revised version of the previous *Miramo Enterprise Module Guide* documentation is split between the *Miramo Reference Guide* (rguide900.pdf) and the new *mmServer Guide* (mmServer900.pdf).
- All jobs are processed via the mmServer Windows service interface (i.e. the miramox interface that used to be part of the **MEM** exclusively). miramo functions as miramox used to function. miramox continues to exist for backwards compatibility only. See [New licensing scheme](#) on page [RN-viii](#).

There is no way to process jobs outside this interface. The result is much faster document processing in the case of short documents and increased overall reliability. It also enables a completely *new licensing scheme* (see [New licensing scheme](#) on page [RN-viii](#)).

- Licensing is no longer based on the number of CPUs or cores in the host machine (see [New licensing scheme](#) on page [RN-viii](#)).

- Miramo 9 includes support for multiple job processing queues, or *processing channel groups* (see [Processing Channels and Processing Channel Groups](#) on page [RN-viii](#)), as well as several other `mmServer` enhancements.
- The following FrameMaker / FrameMaker Server versions are supported: FrameMaker 7.2 through FrameMaker 10.¹

If a user has FrameMaker 7.2 installed, then Miramo 9 operates in 'old compatibility mode', i.e. no direct, internal support for Unicode and other features in FrameMaker 8 and above. When Miramo 9 is run with FrameMaker 8 and above input is expected to be encoded in UTF-8 by default.²

Users with FrameMaker 8 should install Acrobat Professional 9, updated with all available patches, to avoid a bug in Acrobat Distiller that causes Distiller 8 to be unreliable. (This applies regardless of Miramo version.)

- Miramo 9 is functionally and syntactically 99% backwards compatible with Miramo 8.x with the following major exceptions:

Input data streams that include the <Doc> code must be run with a special option '-batch'. If the input data stream includes the <Doc> code and the '-batch' option is not included the job will fail with an error code.

When the <Doc> code is used, a job record is produced for each atomic document or book.³

The '-multi' option is no longer supported. The '-multi' option provides no benefit in the new Miramo 9 document processing architecture.

MMSERVER INTERFACE

All Miramo jobs are processed via the `mmServer` Windows service. `mmServer` manages FrameMaker sessions, ensuring lossless recovery in the event of session failures, and handles job queueing.

The basic features of the `mmServer` interface include the following:

-
1. Also check the Adobe FrameMaker / FrameMaker Server EULAs for license usage conformance.
 2. The `-tENC` option enables the input to be processed as Windows Code Page 1252 or ISO 8859-1, for backwards compatibility. In this case Miramo uses a translation table to translate the text to Miramo's internal UTF-8 encoding when using FrameMaker 8 or above.
 3. The <Doc> code enables processing multiple, distinct documents or books in a single input stream or file. The input may contain 10s, 100s, 1000s or millions of documents, for high speed batch processing. Each document in an input stream is logged separately.

- Expanded mmVisor GUI for configuration and control: processing channels and processing channel groups configuration, start and stop mmServer, as well as specify operating parameters.
- High level APIs for C++, .NET (C#, VB.net, PowerShell) and Java. Documentation for these APIs is provided in standard, hyperlinked HTML format. The APIs expose all functions needed to run jobs, return job record details and control the mmServer service. The mmVisor GUI application is itself written using the published APIs.

The following simple API code examples are taken from the 'Running Miramo' chapter in the *Miramo Reference Guide*.

```
C++  pJobProcessor->createPdf(inputfile, outputfile);
```

```
C#   jobProcessor.createPdf(inputfile, outputfile);
```

```
Java jobProcessor.createPdf(inputfile, outputfile);
```

Special issues:

- Every Miramo install requires a RunAs system user for Miramo jobs, as part of the installation process.

The "RunAs" user name and password must be supplied during installation. If a Windows user of the given name does not exist, a new user is automatically created with appropriate security settings. Otherwise, if a Windows user with the given name and password already exists the appropriate security settings are silently applied.

This has many advantages with regard to security and reliability. Jobs may be run by any user, but they are actually executed by the Miramo Run As user account, i.e. in a secure, controlled environment.

A new command line utility, 'mmaccess', is provided for setting folder access permissions for the RunAs user where necessary.

PROCESSING CHANNELS AND PROCESSING CHANNEL GROUPS

A set of one or more *processing channels* processes jobs in parallel. A *processing channel group* may contain from one to ten processing channels, subject to licensing.

A *processing channel group* is an individual job queue, referenced either by an administrator-assignable name or a sequence number (1, 2, 3, etc). Processing channels and processing channel groups are managed by the mmServer Windows service.¹

The following examples assume a multi-CPU, multi-core server.

- Example 1.** All ten processing channels are assigned to the default processing channel group, say "Accounts". All jobs are processed in order. Up to 10 jobs may run in parallel. If more than 10 jobs are requested, the extra jobs will wait in the queue until a processing channel is free.
- Example 2.** Four processing channels are assigned to "Accounts", three processing channels are assigned to "Marketing", two to "Catalogs", etc. If "Accounts" runs five jobs at once, the fifth job will wait in a queue until one of the four "Accounts" processing channels becomes free. And so on. Another possible configuration is illustrated in Figure RN-1.

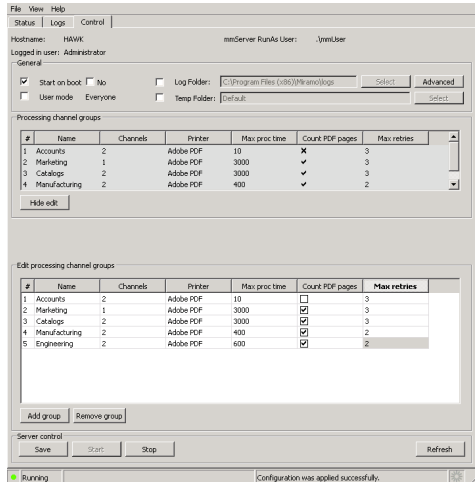


Figure RN-1 Processing channel groups

A Miramo '-processGroup xxxx' option (or <MiramoXML processGroup="xxxx" > attribute) may be used to specify which processing channel group should run a job. The value of xxxx may be either the integer value of the processing channel group (from 1 upwards) or its assigned name. If Miramo is run without the '-processGroup' option then jobs are run by the default processing channel group (1)—thus backwards compatibility with existing applications is preserved.

NEW LICENSING SCHEME

Licensing and pricing is no longer dependent on the number of CPUs and cores in the host machine. Instead license categories are based on the number of processing channels.

There are three commercial license types:

- Single processing channel license (1PC).* This is the entry-level license.

1. If a *processing channel group* contains more than one *processing channel* then the jobs assigned to the processing channel group are processed in parallel by the processing channels in the group.

In this scenario only one job can be processed at a time. If additional jobs are requested before the first job has completed, the later jobs are queued and processed in sequence.

- *Dual processing channel license (2PC)*. This license is for a maximum of two processing channels in a single processing channel group, or two processing channel groups, each with a maximum of one processing channel.

In the case of a *Dual processing channel license (2PC)* two jobs can be processed concurrently. If additional jobs are requested when two other jobs are already running, the later jobs are queued and processed in sequence.

- *Unlimited processing channels license (UPC)*. This license will be for any number of processing channels and processing channel groups, generally up to a maximum of 10.

License types '1' and '2', **1PC** and **2PC**, are designed for small scale environments having low average and low peak document production loads. The **UPC** license type is designed for enterprise environments requiring fast turnarounds, high volume throughputs and no bottlenecks.

For license types '1' through '3', **1PC**, **2PC** and **UPC**, a host-bound license activation key is needed.

UPGRADES

Upgrades from previous Unlimited CPU licenses to the new Unlimited Processing Channels (**UPC**) license type is straightforward and is on the same basis as previously.

Upgrades from other current license types is more complicated, owing to the changes in the Miramo 9 licensing model—in some cases there is no exact correspondence between old and new license types.

A specific upgrade price list addresses this issue. Upgrade pricing depends on the following: the Miramo version that will be upgraded, the type of Miramo 9 license being upgraded to and whether or not Miramo Maintenance is in effect at the time of upgrade.

NEW FEATURES (v. 9.2)

The following is a summary of new features in Miramo 9.2.

- Tables with proportional-width columns may be included within a table cell, or a set of horizontally spanned cells, and have its width auto-adjusted to the overall width of the container(s). This applies even when the container

cell(s) are specified with proportional-width columns. Multiple table-within-table nesting levels are supported.

See the description of the `<Tbl ... >` code `'tblWith'` auto setting on page [R-252](#) in the *Miramo Reference Guide*. Backwards compatibility has been maintained.

The calculation rules used for determining the widths of proportional-column width tables within table cells are described in section [Auto-sizing tables to width of table cells](#) on pages [R-273–275](#) in the *Miramo Reference Guide*.

- When a multi-page PDF file is imported the size of each page in the PDF file is used to determine the default dimensions of the page in the output document. Previously all PDF pages in an imported PDF file were assumed to have the same dimensions as the first page. See [PDF file import](#) on pages [R-97–98](#).
- The 2 GB limit on the size of intermediate .tps files has been removed.
- Improved internationalization support.
- The `mmUser RunAs` user is a no-logon user by default. This can be changed using the `mmServer '-nologon'` command-line option, described on page [MS-17](#) in the *mmServer Guide*.

NEW FEATURES (v. 9.1)

The following is a summary of new features in Miramo 9.1.

- The `<PostProcess>` and `<Processor ... />` codes enable fine control over the invocation timing of one or more user-created FDK APIs, `ExtendScripts` and `FrameScripts` on an in-document and a book-wide basis via the `<Processor ... />` `'applyWhen'` and `'applyTo'` options. See pages [R-374–384](#) in the *Miramo Reference Guide*.
- *Cache images for use in multiple jobs.* By default, when the `<Image ... />` `'imode'` option is set to `bypass`, all images in TIFF and PNG format are cached for optional re-use within jobs. This results in faster throughput.

The new `'imageCache'` command line and `<MiramoXML ... >` option enables preserving the image cache between jobs. See pages [R-9](#) and [R-458](#) in the *Miramo Reference Guide*.

- Enhancements to `mmServer` and `mmVisor`
 - Improved diagnostics in the event of a `mmServer` initialization failure. Special key combination to start `FrameMaker` as `RunAs` user: `Ctrl-Alt-f`.¹

- Apply product license key via the mmVisor Help menu.
- Edit and save-to-new PDF .joboptions files for use by the Miramo RunAs user.
- mmVisor optionally displays user account name of the job initiator. This information is also stored in the job record.
- mmVisor enables configuration of network shares for access by the Miramo RunAs user.
- mmVisor shows tool-tip descriptions for an extended set of numeric exit codes.
- The file mmvisor910setup.exe is a self-contained installer that enables installation of mmVisor, rmmcmd and .dlls for supporting C++, Java and .NET APIs on hosts that do not have Miramo installed, for remote job processing and monitoring. This file is contained in the folder:
%MM_HOME%\files.
- Logon access to the Miramo RunAs user account may be enabled/disabled using the mmServer 'noLogon' command line option. See page [MS-17](#) in the *mmServer Guide*.

As a general rule, no one should ever logon to the mmServer RunAs user account.

NEW FEATURES (v. 9.0)

The following is a summary of the major additional new features in Miramo 9.

- Support for PNG and TIFF image file formats when the <Image ... /> code 'imode' option is set to bypass, which can greatly improve performance. See page [R-180](#) in the *Miramo Reference Guide*.

1. This feature is intended for diagnostics only, not for creating templates.

- Support for color blends / gradients as background fills in a range of dynamically-sized graphic objects, e.g. single-page tables, table rows, table cells, text frames and anchored text frames in PDF output, as illustrated in Figure RN-2.

By default the above background fills apply to the entire object area. In addition background fills may have the following shapes: ellipse, circle, diamond, rounded rectangle (with individually specifiable corner radii). These shapes are sized dynamically in relation to the dimensions of the fill area.

The features are attributes of the `<FrameFill ... />` code, described on pages R-148–158 in the *Miramo Reference Guide*.

- Support for *alpha-channel transparency* in images in PDF output. Previously the `tpllmode -bypass` and the corresponding `<Image>` code `imode` option supported EPS and JPEG image file formats. This is extended to support PNG and TIFF files, including the correct handling of alpha-channel data. Partially transparent background tints are also supported.
- Image transparency in the range 0% to 100% may be applied to EPS, JPEG, PNG and TIFF images, with or without alpha-channels, as well as to all `<FrameFill ... />` color fills. Multiple different blend modes may be specified. See page R-185 in the *Miramo Reference Guide*.
- Support for C++, Java and .NET job processing and monitoring APIs.
- Expanded support for the CMYK color model.¹
- The text background-color feature of FrameMaker 10 and above is also supported. See page R-141 in the *Miramo Reference Guide*.

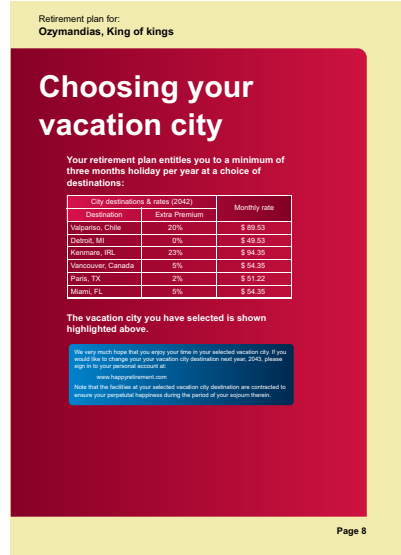


Figure RN-2 Gradients & transparency. (The input to produce the output illustrated above is shown in Example 3.51 on page R-157).

1. Requires FrameMaker 10 or above.

APPENDIX—PROCESSING CHANNEL THROUGHPUT

DESKTOP RELATIVE PERFORMANCE

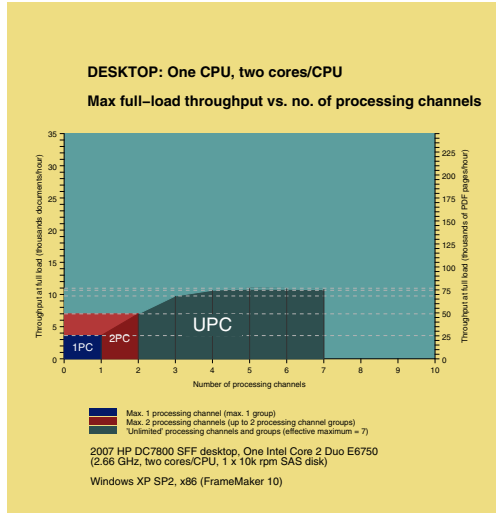


Figure RN-3 Relative desktop performance scaling

SERVER RELATIVE PERFORMANCE

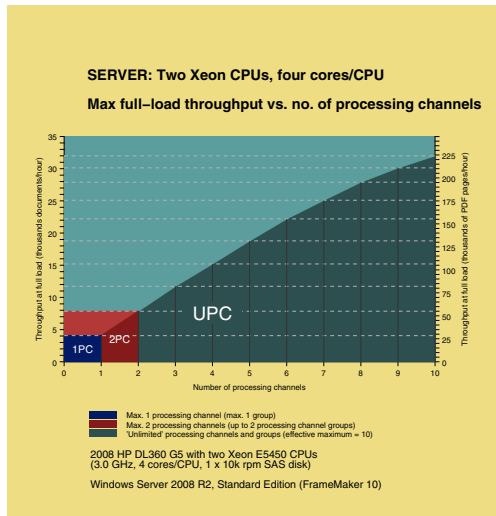


Figure RN-4 Relative server performance scaling

