



FastReport for Xailer and [x]Harbour

©2019 OZ Software

<http://www.ozs.com>

Guía del Usuario

por Ignacio Ortiz de Zúñiga

xaConta

*Copyright OZ Software 2019
Todos los derechos reservados*

FastReport for Xailer and [x]Harbour

© 2019 OZ Software

Reservados todos los derechos. El nombre de producto xaConta y su logo son marcas comerciales registradas de OZ Software. Otras marcas y nombres de productos son marcas comerciales o registradas de sus respectivos propietarios.

Ninguna parte de esta documentación puede copiarse, fotocopiarse, reproducirse, traducirse, microfilmarse o duplicarse por cualquier medio sin el consentimiento previo por escrito de OZ Software.

Impreso en Madrid en: enero 2019.

Agradecimientos:

A todos los usuarios que han participado en la fase Beta del producto aportandonos valiosa información sobre errores y sugiriendo mejoras que han podido incorporarse al producto.

Por último, a todos los usuarios de OZ Software que nos han apoyado y animado desde hace muchos años.

Tabla de Contenido

Prefacio	0
Capítulo I FastReport for [x]Harbour	9
1 Linking & Registration	12
2 Whatsnew	12
3 License	13
4 Xailer vs. Standard version	18
5 T Classes and X classes	18
6 Sergey Spirin compatibility	19
7 TFastReport	20
TFastReport:Properties	20
TFastReport:aDatasets.....	20
TFastReport:cDllPath.....	21
TFastReport:cFileName.....	21
TFastReport:cFRLicense.....	21
TFastReport:cLanguage.....	22
TFastReport:cLastError.....	22
TFastReport:cResIco	22
TFastReport:cResName	22
TFastReport:cTitle.....	23
TFastReport:cXALicense	23
TFastReport:IAbortOnErrors	23
TFastReport:nLastError.....	23
TFastReport:oEngineOptions.....	24
TFastReport:oPreviewOptions.....	24
TFastReport:oPrintOptions	24
TFastReport:oReportOptions	24
TFastReport:Methods	25
TFastReport:AddArray	26
TFastReport:AddCategory	26
TFastReport:AddDbf.....	26
TFastReport:AddFunction	27
TFastReport:AddVariable	27
TFastReport:Calc	28
TFastReport:CategoriesList	28
TFastReport:Clear	28
TFastReport:ClearMasterDetail.....	28
TFastReport:ClearVariables	29
TFastReport>Create.....	29
TFastReport:DelDataset	29
TFastReport:DelError	30
TFastReport>DeleteCategory	30
TFastReport>DeleteVariable	30
TFastReport:DesignReport	30
TFastReport:DoExport	31
TFastReport:GetErrors	31
TFastReport:GetProperty	31

TFastReport:GetVariable	32
TFastReport:LoadFPFile.....	32
TFastReport:LoadFromFile	33
TFastReport:LoadFromResource.....	33
TFastReport:LoadFromString.....	33
TFastReport:LoadLangFile.....	33
TFastReport:LoadStyleSheetFromFile.....	34
TFastReport:LoadStyleSheetFromString.....	34
TFastReport:New	34
TFastReport:PrepareReport	35
TFastReport:PreviewClose.....	35
TFastReport:Print.....	35
TFastReport:SaveToFile.....	35
TFastReport:SaveToFPFile.....	36
TFastReport:SaveToString.....	36
TFastReport:SetADOConnectStr.....	36
TFastReport:SetFormatSettings	36
TFastReport:SetIcon.....	39
TFastReport:SetLicense	39
TFastReport:SetProperty.....	39
TFastReport:SetText	40
TFastReport:SetVariable.....	40
TFastReport>ShowReport.....	40
TFastReport:VariablesList	41
TFastReport:Events	41
TFastReport:bOnAfterLoad	41
TFastReport:bOnAfterPrint.....	42
TFastReport:bOnAfterPrintReport	42
TFastReport:bOnBeforeConnect	42
TFastReport:bOnBeforeCreate.....	42
TFastReport:bOnBeforeLoad	43
TFastReport:bOnBeforePrint.....	43
TFastReport:bOnBeginDoc	43
TFastReport:bOnBtnGenPdf.....	43
TFastReport:bOnBtnPrint.....	43
TFastReport:bOnBtnSendMail.....	44
TFastReport:bOnClickObject	44
TFastReport:bOnEndDoc	44
TFastReport:bOnEndPreview	44
TFastReport:bOnGetValue	45
TFastReport:bOnMouseOverObject	45
TFastReport:bOnPageChanged	45
TFastReport:bOnPreview	45
TFastReport:bOnPrintPage.....	46
TFastReport:bOnPrintReport	46
TFastReport:bOnProgress	46
TFastReport:bOnProgressStart	46
TFastReport:bOnProgressStop.....	47
8 TFrDataset	47
TFrDataset:Properties	47
TFrDataset:aRelationFields.....	48
TFrDataset:cName.....	48
TFrDataset:nLoaded.....	48
TFrDataset:nMaxRecsOnDesign	49

TFrDataset:oDsMaster.....	49
TFrDataset:oReport.....	49
TFrDataset:Methods	49
TFrDataset:IsActive.....	50
TFrDataset:IsLoaded.....	50
TFrDataset:Refresh.....	50
TFrDataset:SetMaster.....	50
TFrDataset:Events	51
TFrDataset:bOnAfterLoad	51
TFrDataset:bOnClose	51
TFrDataset:bOnFirst	51
TFrDataset:bOnNext	51
TFrDataset:bOnOpen	52
TFrDataset:bOnPrior	52
9 TFrDbfDataset	52
TFrDbfDataset:Properties	52
TFrDbfDataset:aFields	52
TFrDbfDataset:ILoadOnDemand	53
10 TFrArrayDataset	53
TFrArrayDataset:Properties	53
TFrArrayDataset:aData	54
TFrArrayDataset:aFields	54
11 TFrEngineOptions	54
TFrEngineOptions:Properties	54
TFrEngineOptions:cTempDir.....	55
TFrEngineOptionsn:IConvertNulls	55
TFrEngineOptions:IDoublePass	55
TFrEngineOptions:IgnoreDevByZero	55
TFrEngineOptions:IPrintIfEmpty	55
TFrEngineOptions:ISilentMode	56
TFrEngineOptions:IUseFileCache	56
TFrEngineOptions:nMaxMemSize.....	56
TFrEngineOptions:nNewSilentMode	56
12 TFrPreviewOptions	57
TFrPreviewOptions:Properties	57
TFrPreviewOptions:IAccelerate	57
TFrPreviewOptions:IDoubleBuffered	58
TFrPreviewOptions:IMaximized	58
TFrPreviewOptions:IModal	58
TFrPreviewOptions:IOutlineExpand	58
TFrPreviewOptions:IOutlineVisible	58
TFrPreviewOptions:IPictureCacheInFile	59
TFrPreviewOptions:IShowCaptions	59
TFrPreviewOptions:nButtons	59
TFrPreviewOptions:nOutlineWidth	60
TFrPreviewOptions:nZoom	60
TFrPreviewOptions:Methods	60
TFrPreviewOptions:SetBounds	60
13 TFrPrintOptions	61
TFrPrintOptions:Properties	61
TFrPrintOptions:cPageNumbers	61
TFrPrintOptions:cPrinter	61

TFrPrintOptions::ICollate.....	62
TFrPrintOptions::IReverse	62
TFrPrintOptions::IShowDialog	62
TFrPrintOptions::nCopies.....	62
TFrPrintOptions::nPrintPages.....	62
TFrPrintOptions::Methods	63
TFrPrintOptions::ClearOptions.....	63
14 TFrReportOptions	63
TFrReportOptions::Properties	63
TFrReportOptions::aDescription.....	64
TFrReportOptions::cAuthor.....	64
TFrReportOptions::cInitString.....	64
TFrReportOptions::cPassword.....	64
TFrReportOptions::lCompressed.....	64
Índice	66

Capítulo

I

1 FastReport for [x]Harbour

Introduction

FastReport (FR) is a visual report generator that provides to your applications the ability to generate multiple types of reports quickly and efficiently.

This report generator provides all the tools necessary to develop complex and showy reports.

FastReport for [x]Harbour is based on FastReport from Fast Report Inc. <http://www.fast-report.com/> that also has versions for Delphi (VCL and FireMonkey), .NET and Mono

The first adaptation to [x]Harbour compiler was done by Sergey Spirin from ParitetSoft. Unfortunately that product id discontinued. Xailer decided to create its own adaptation trying to maintain the compatibility with the old ParitetSoft product, in order to minimize as much as possible the adaptations of old reports.

FastReport can do all kind of reports, from columnar reports, labels, graphics, bar codes, calendars, crosstabs reports, invoices, large tickets, photo and rich text printing.

Any kind of printing on any printer can be done with FastReport.

FastReport features

- No complicated installation, only a royalty-free DLL is needed on your application directory.
- Powerful visual designer, multilingual and easy to use.
- Completely WYSIWYG
- Band design oriented.
- Supports any type of components: lines, boxes, texts, images, OLE, graphics, bar codes, rich text, etcetera.
- Export functionality with multiple formats available (PDF, HTML, RTF, CSV, TXT, XLS, XML, JPG, BMP, TIFF, GIF, ODS, ODT, eMail), from the preview window or directly on your source code without showing anything on the screen.
- Support for any kind of printer, including dot matrix printing.
- Internal script editor with syntax highlighting.
- Support of various scripting languages: PascalScript, C++Script, Javascript and BasicScript.
- Scripting debugger.
- Runtime access of any Xailer variable, function or work area.
- Great speed on report generation and export.
- Third party libraries to expand its possibilities. For example QR bidimensional codes.
- Report preview with search and edit capabilities.
- Text rotation.

- Paragraphs justification.
- Support of simple HTML tags on text objects (Font,color,b,i,u,sub,sup).
- Multiple types of fonts, sizes, styles and colors.
Note: You may use any font on the developing process, but that font must exist on the end user PC, in case not, a font substitute will be used.
- Word wrapping.
- Use of functional URLs (in preview mode).
- Zoom on the visual designer.
- Guides and rules.
- Configurable grid in mm, inches or pixels.
- Wizard for most common reports.
- Object copy to Windows clipboard.
- Unlimited Undo/Redo.
- The reports can contain their own forms, for example, to ask for some parameters. You may have all the forms you need. FastReport supports the common objects used on any form designer: buttons, text edit, check boxes, etcetera.
- Los informes pueden contener sus propios formularios, por ejemplo para pedir parámetros antes de preparar un informe. Se pueden tener tantos diálogos como sea necesario. FastReport usa el mismo diseñador para crear y mantener los diálogos y soporta los controles de Windows mas comúnmente usados como botones, gets, check boxes, etc.
- Storage of all the reports in XML format. Compressed (GZip compatible) also supported..
- Bidirectional text output for specific languages.
- Support of third party components through OLE, like Excel files for example.
- Any kind of page dimensions and even with unlimited length.
- Includes a full functional demo with the only limitation that only prints the first five pages.

FastReport components

FastReport has four main components:

- A dynamic link library **FRX.DLL** that includes the FastReport engine, including all the languages that may use. The library can be copied to any directory pointed out by your PATH environment variable. But normally the best place is to copy it to the same directory of your application.
- **TFastReport** main class to manage the reports.
- **TFrDataset** and its descendants: **TFrArrayDataset** and **TFrDbfDataset** are classes which its purpose it to connect the the different source datasets with FastReport itself.
- **TFROptions** and its descendants: **TFrEngineOptions**, **TFrPreviewOptions**, **TFrPrintOptions**

and **TFrReportOptions** are classes that maybe used to get or set all the properties of any report.

Using the Report Generator

FastReport is not intended to be used as normal application, is not an EXE file, is a report generator that is embedded into your program.

To use it you just need to include the following code in your application:

```
oFr:=TFastReport () :New ()
oFr:DesignReport ()
oFr:End ()
```

When running your application and executing this code, the report designer will be shown with all its features so you will be able to design your own report and then save it with any name, for example "Sales report.Fr3" which is a XML file that we can open with any text editor in case necessary.

In order to run a already created report to a preview window or directly to the printer, a few lines more are necessary:

```
oFr:=TFastReport () :New ()
oFr:LoadFromFile("Sales report.fr3")
oFr>ShowReport ()
oFr:End ()
```

Really simple.

Using the Report Designer

The use of FastReport is really simple and there is a lot of documentation available. For example:

http://www.fast-report.com/en/download/public_files/142/
http://www.fast-report.com/en/download/public_files/146/
http://www.fast-report.com/en/download/public_files/148/

For further information consult the following link:

<http://www.fast-report.com>

Acknowledgements

We would like to thanks Bingen Ugaldebere and René Flores for its great job on the Spanish documentation of FastReport and for allow us to use part of their work on the Spanish version of this manual.

Likewise, be grateful with Sergey Spirin (RIP) for its work on making FastReport accesible to a lot of Harbour users. We collect its witness and expect to fulfill the expectations of all the [x]Harbour community.

1.1 Linking & Registration

All the modules needed to access the FRX.DLL are included in Xailer libraries (from version 3.1 and above), so you will not need nothing else. In case you want to use FastReport for [x]Harbour apart from Xailer with any other GUI framework or even in console mode, then you will need to link your programs with a small library.

Xailer for [x]Harbour includes different libraries depending on the compiler used:

- **libxaFrhMin.a**: For Harbour and MinGW
- **xaFrhBcc.lib**: For Harbour and Borland C++
- **xaFrxBcc.lib**: For xHarbour and Borland C++
- **xaFrXPelles.lib**: For xHarbour and Pelles C (xHarbour.com)

From version 1.2 and above the library '**version.lib**' ('**libversion.a**' on MinGW) is needed in order to check for the correct DLL version so it must be also linked. This library is just an imported library from the Windows system DLL: VERSION.DLL.

In order to stop seeing the annoying watermark and the five pages limitation you must register your version of FastReport for [x]Harbour & Xailer on every application you may create. This can be done by setting the properties TFastReport:cFRLicense and TFasReport:cXALicense every time you create a TFastReport object. This can be a heavy task that can be resolved with just a few lines of code on any of your source code modules. Just include the following code:

```
#include "HbClass.ch"

#pragma TEXTIDDEN(1)
CLASS TFastReport FROM XFastReport

    VAR cFRLicense INIT "1234-5678-9012-3456" // your FR License goes
here
    VAR cXALicense INIT "XAFR-00000000-01"      // your XA License goes
here
END CLASS
#pragma TEXTIDDEN(0)
```

The TEXTIDDEN commands permit to obfuscate your license on the executable. Without those lines your license can be easily accessed. We highly appreciate your cooperation on this issue.

If you want to understand how this code works take a look at the chapter T Classes and X Classes.

1.2 Whatsnew

Fixes up to current version 1.5.11.27):

- Updated to FastReport VCL 4.15

Fixes up to current version 1.4.11.19):

- Own FastReport preview pages disappear when overloading export preview buttons and calling the standard DoExport() method

Fixes up to current version 1.4.11.3):

- Control of incorrect fields in SetMasterDetail() calls
- Error on PrepareReport() and direct printing
- Error when reloading a already loaded dataset
- Error on accesing TFrReport:oFont object with method SetProperty

From version 1.3 to 1.4:

- QR code support
- Empty datasets were not available at design time
- AddVariable always returned true even if failed
- New methods: ClearVariables, VariablesList and CategoriesList
- New method: TFastReport:SetFormatSettings
- New method: TFastReport:PreviewClose

From version 1.2 to 1.3:

- New property TFrxDbfDataset:ILoadOnDemand to load data on demand
- Error on method SetIcon()
- Error with ADS custom field types
- Overloaded FRReportManager:AddVariable() to accept the same parameters than Sergey software

From version 1.1 to 1.2:

- Harbour version compiled with latest version of Harbour 3.2.0dev (r1310281618)
- MinGW+Harbour version compiled with latest version of MinGW 4.8.1
- New property TFastReport:cDIIPath
- New event TFastReport:OnBeforeCreate
- Error on method TFastReport:LoadFromResource
- The function CallHBFunc() can now be called with no parameters
- New function support GetHbArrayVar

From version 1.0 to 1.1:

- New events
- New method PrepareReport
- Error on method TFasReport:DoExport
- Error on method TFastReport:SetProperty

1.3 License

SINGLE USER SOFTWARE LICENSE AND LIMITED WARRANTY

PLEASE READ THIS SOFTWARE LICENSE AGREEMENT CAREFULLY BEFORE USING THE SOFTWARE. BY USING THIS PRODUCT, YOU ARE CONSENTING TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, RETURN THE PRODUCT TO THE PLACE OF PURCHASE FOR A FULL REFUND.

This license agreement covers your use of the FastReport for Xailer & [x]Harbour, its source code,

documentation, and executable files, hereinafter referred to as "Product". The Product is Copyright © 2013 Xailer.com & Ramón Argüelles. You may use it and distribute it according to this following License Agreement. If you do not agree with these terms, please remove the Product from your system. By incorporating the Product in your work or distributing the Product to others you implicitly agree to these license terms.

The Product is, and remains, Copyright © 2013 Xailer.com & Ramón Argüelles, with exception of specific copyrights as noted in the individual source files.

The Product is distributed as try-before-you-buy product. This means:

All copyrights to Product are exclusively owned by the author - Xailer.com & Ramón Argüelles. Product is protected by copyright laws. At all times Xailer.com & Ramón Argüelles retains full title to the software. Subject to your acceptance of and accordance with the terms and conditions stated in this agreement, you shall be granted a single-user software license. Any previous agreement with Xailer.com & Ramón Argüelles is superseded by this agreement.

Anyone may use trial version of Product as long as you want. You can try trial version of Product, which prints only five page of report. Full version of Product not contains such restrictions. To use full version of Product, you MUST register.

The Product unregistered trial version, may be freely distributed, with exceptions noted below, provided the distribution package is not modified. No person or company may charge a fee for the distribution of Product without written permission from the copyright holder. The Product unregistered trial version may not be bundled or distributed with any other package without written permission of the copyright holder.

Xailer Inc. reserves the right to monitor proper use of this software, invalidating serial numbers in the event of misuse and offering to rightful owner's new serial numbers if necessary.

REGISTER THIS SOFTWARE LICENSE GIVES YOU THE RIGHT TO:

1. Install and use the Product for the sole purposes of designing, developing, testing, and deploying application programs which you create. You may install a copy of the Product on a computer and freely move the Product from one computer to another, provided that you are the only individual using the Product. If you are an entity, you must designate one individual within your organization ("Named User") to have the right to use the Product.
2. Write and compile your own application programs using the Product contained in this package. All copies of the Product you write and distribute must include a Xailer.com & Ramón Argüelles copyright notice, or a valid copyright notice of your own.
3. Make one copy of the Product for backup or archival purposes or copy the Product to a single permanent storage medium provided you keep the original solely for backup or archival purposes.
4. Technical support and notifications on Product updates. Free upgrades for one year from the date of purchase.
5. The registered Product may not be rented or leased, but may be permanently transferred, if the person receiving it agrees to terms of this license. If the software is an update, the transfer must include the update and all previous versions.
6. Along with our license will also receive an original license from FastReport Company that will be required for product registration.
7. It is not provided any additional license deductions, except for cost of the registration, connected

with creation and distribution of reports and forms of Product. The registered users, can use Product as "Royalty free". It means, that they freely may distribute the programs using Product if it does not contradict conditions of this license agreement. Any sanctions to that on from the author is not required.

8. This product includes a FastReport 5 VCL Standard edition license.

ENGAGING IN ANY OF THE ACTIVITIES LISTED BELOW WILL TERMINATE THE SOFTWARE LICENSE. IN ADDITION TO SOFTWARE LICENSE TERMINATION, XAILER.COM. MAY PURSUE CRIMINAL, CIVIL, OR ANY OTHER AVAILABLE REMEDIES.

1. Distribution of any files contained in this software package, other than the runtime library (dll) as a part of users applications.
2. Modification, decompilation, disassembly, reverse engineering or translation of the Product.
3. Removal of proprietary notices, labels or marks from the Product or Product Documentation.
4. Inclusion of the Product in a development environment (apart from Xailer), CRM or ERP systems.
5. Creation of an application that does not differ materially from the Product.
6. Development and/or distribution of a stand-alone reporting application based on the Product.
7. Creation of an application (whether it be freeware, shareware or a commercial product) which competes directly or indirectly with the Product.

In these cases you need other type of license.

AGREEMENT PERTAINING TO THE RELEASE OF SOURCE CODE BY XAILER.COM & RAMON ARGÜELLES TO RECIPIENT:

USE OF SOURCE CODE

Recipient will not utilize the source for the creation of Product (whether it be freeware, shareware or a commercial product) which competes directly or indirectly with Product. In addition, Recipient will not disclose the source itself, nor the implementations discovered therein, to any party involved in the creation of software which competes directly or indirectly with Product.

DISTRIBUTION OF SOURCE CODE

Recipient will not distribute the Product. Specifically this includes all prg, ch files which Xailer.com has provided.

CHANGES TO SOURCE CODE

Xailer.com & Ramón Argüelles have the right to change any part of the source in future versions of the Product. These changes may include the removal of classes, properties and methods or the creation of new classes, properties and methods.

TECHNICAL SUPPORT FOR SOURCE CODE

Xailer.com will not provide support for changes Recipient makes to the source. Recipient assumes full responsibility for supporting any code or application which results from such modification. Recipient will not hold Xailer.com liable, directly or indirectly, for any changes made to the source, including changes which Recipient has made based on advice or suggestions provided by Xailer.

com.

You must clearly indicate any modifications at the start of each source file. The user of any modified Product code must know that the source file is not original.

SOURCE IS PROVIDED AS IS PRODUCT IS DISTRIBUTED "AS IS". NO WARRANTY OF ANY KIND IS EXPRESSED OR IMPLIED. YOU USE AT YOUR OWN RISK. XAILER.COM WILL NOT BE LIABLE FOR DATA LOSS, DAMAGES, LOSS OF PROFITS OR ANY OTHER KIND OF LOSS WHILE USING OR MISUSING THIS PRODUCT

TERM AND TERMINATION

The license granted under this Agreement will continue in force until terminated, as set forth herein. If Licensee fails to pay any monies or provide any services due in connection with the Product, or violates any term or condition of this Agreement, Xailer.com or its agent may terminate this License immediately by giving notice of termination to Licensee. Licensee is responsible for providing valid contact information to Xailer.com. If no valid contact information is available for Licensee in Xailer.com records, Xailer.com is not required to give notice of termination to Licensee. Licensee also may terminate this License voluntarily by giving notice of termination to Xailer.com and destroying or returning to Xailer.com all copies of all or any part of the Product and related user documentation in Licensee's possession or under Licensee's control.

EFFECT OF TERMINATION

Immediately upon termination, Licensee will destroy or return to Xailer.com all copies of all or any part of the Product in Licensee's possession or under Licensee's control.

Licensee will have no right to keep or use any copy of the Product and related user documentation for any purpose after termination of this Agreement.

TRANSFER OF PRODUCT

Licensee shall not have the right to transfer this Product license, without the prior written consent of Xailer.com.

CONFIDENTIALITY.

The parties to this Agreement will take all reasonable steps to ensure that any material or information identified by either party to be confidential ("Confidential Information"), which the other party has possession or knowledge of in connection with this Agreement, will not be disclosed to others, in whole or in part, without the prior written permission of the other party. Neither party will have the obligation to maintain the confidentiality of any data or information which (i) was in the receiving party's lawful possession prior to receipt from the other party, (ii) is later lawfully obtained by the receiving party from a third party having no obligation of secrecy to the other party, (iii) is available to the public through no act or failure of the receiving party, (iv) is readily available in the public domain, or (v) is independently developed by the receiving party. The receiving party will immediately return or destroy any or all Confidential Information that has been provided to it by the other party, upon the other party's request.

PRODUCT WARRANTY

Except with respect to the Redistributables, which are provided "as is," without warranty of any kind, Xailer warrants that the Software, as updated and when properly used, will perform substantially in accordance with the accompanying documentation, and the Software media will be free from defects in materials and workmanship, for a period of ninety (90) days from the date of receipt. Any implied warranties on the Software are limited to ninety (90) days. Some states/jurisdictions do not allow limitations on duration of an implied warranty, so the above limitation may not apply to you.

REFUNDS

In the event that Xailer.com refunds any amounts paid by Licensee for the Product, pursuant to Paragraph above, Licensee understands and agrees that this Agreement and the Product license are terminated, and Xailer.com will withdraw Product, code and related documentation provided under this Agreement. Upon receipt of such refund, Licensee agrees that it will no longer use, display or otherwise control the Product, code or related documentation for which the refund was issued.

DISCLAIMER OF WARRANTY FOR PRODUCT EXCEPT FOR THE EXPRESS WARRANTIES SET FORTH IN THIS AGREEMENT, XAILER.COM DISCLAIMS ALL IMPLIED WARRANTIES FOR THE PRODUCT, INCLUDING WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. XAILER.COM MAKES NO REPRESENTATIONS CONCERNING THE QUALITY OF THE PRODUCT AND DOES NOT PROMISE THAT THE PRODUCT WILL BE ERROR FREE OR WILL OPERATE WITHOUT INTERRUPTION.

LIMITATION OF LIABILITY

IN NO EVENT WILL XAILER.COM BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR OTHER DAMAGES ARISING OUT OF THE USE OF THE PRODUCT BY ANY PERSON, REGARDLESS OF WHETHER FAST REPORTS IS INFORMED OF THE POSSIBILITY OF DAMAGES IN ADVANCE. THESE LIMITATIONS APPLY TO ALL CAUSES OF ACTION, INCLUDING BREACH OF CONTRACT, BREACH OF WARRANTY, XAILER.COM' NEGLIGENCE, STRICT LIABILITY, MISREPRESENTATION AND OTHER TORTS.

OWNERSHIP OF PRODUCT

Xailer.com has and will retain all ownership rights in the Product, including all patent rights, copyrights, trade secrets, trademarks, service marks, related goodwill and confidential and proprietary information. Licensee will have no rights in the Product except as explicitly stated in this Agreement.

ASSIGNMENT AND DELEGATION

Licensee may not assign this Agreement or any rights under it and may not delegate any duties under this Agreement without Xailer.com' prior written consent. Any attempt to assign or delegate without that consent will be void.

GENERAL

This Agreement constitutes the entire understanding between Xailer.com and Licensee with respect to subject matter hereof. Any change to this Agreement must be in writing, signed by Xailer.com and Licensee. Terms and conditions set forth in any purchase order which differ from, conflict with, or are not included in this Agreement, shall not become part of this Agreement unless specifically accepted by Xailer.com in writing.

Licensee shall be responsible for and shall pay, and shall reimburse Xailer.com on request if Xailer.com is required to pay, any sales, use, value added (VAT), consumption or other tax (excluding any tax that is based on Xailer.com' net income), assessment, duty, tariff, or other fee or charge of any kind or nature that is levied or imposed by any governmental authority on the Product.

All rights not expressly granted here are reserved by Xailer.com & Ramón Argüelles.

LICENSEE HAS READ THIS AGREEMENT AND UNDERSTANDS AND AGREES TO ALL OF ITS TERMS AND CONDITIONS.

Thank you for using FastReport for Xailer & [x]Harbour!
Xailer.com.

1.4 Xailer vs. Standard version

There is a single product FasReport for [x]Harbour & Xailer, not two. You may use it on your Xailer applications or with any other [X]Harbour GUI or even in console mode. The license gives you the gran to use it with any [x]Harbour framework.

The Xailer version of FastReport for [x]Harbour has some important features that are not present on the standard version, which are:

- Full integration with Xailer's datacontrols system: Within Xailer there is a new component **TFrXailerDataset** which makes the connection between FastReport datasets and Xailer datasets
- Visual design: All the design is completely visual from Xailer's IDE.
- Preview reports inside Xailer forms. There is a special TFrPreviewControl for that purpose.
- Use of events: Xailer includes a special type of class members with the name 'Event' which resembles to code-blocks but is much more powerful. On the standard versin of FR for [x] Harbour all events members have been turned into code-blocks, but in this documentation are treated like events.

Xailer and standard versions of FastReport for [x]Harbour use the same DLL.

In this documentation you will constantly see references to the Xailer version of FR for [x]Harbour, never mind, they are alo true for the standard version of FR for [x]Harbour, except when explicitly said.

1.5 T Classes and X classes

This is something almost every Xailer user knows what is about, but regular [x]Harbour users may not. Xailer uses a two phase class container for any class:

- Instanciable objects may only use the 'T' prefixed classes. That is to say: You can only instantiate 'T' classes.
- 'X' classes hold all the complexity while 'T' classes are just empty classes that inherit from its 'X' class.

For example, this is the hierarchy of **TFrPrintOptions** class:

```
TFrPrintOptions
  XFrPrintOptions
    TFrOptions
      XFrOptions
```

```
TFrObject
XFrObject
```

And this is the code of TClasses:

```
CLASS TFrPrintOptions FROM XFrPrintOptions
END CLASS
```

```
CLASS TFrOptions FROM XFrOptions
END CLASS
```

```
CLASS TFrObject FROM XFrObject
END CLASS
```

As you can see there are completely empty. There is no code on them. The reason for this is simple. If you want to change the behaviour of any class you do not need to inherit from TFrPrintOptions you just need to overload it completely, that's all. This way you do not need to change a single line of your code to point to a new class.

Take a look at property TFastReport:cFRLicense for more information.

1.6 Sergey Spirin compatibility

FastReport for [x]Harbour has made a great effort in order to be the maximum compatible with the Sergey Spirin product, but still there are minor differences in subjects we believe are not crucial, like:

- Specific BLOB methods since our version does not need any special treatment for them
- SetEventHandler method since we use a better approach with specific events
- Resync method since is absolutely not necessary on our system
- SetTxtDataset method since we believe is useless
- Multiple reports mode
- Manual build reports

Our base class is TFastReport but you can also instantiate your objects from **frReportManager()** which adds to TFastReport some compatibility methods with Sergey product. Which are:

- Constructor **New(cFilename)**
- Constructor **Init()**
- Method **DestroyFR()**
- Method **SetWorkArea(cAlias, nArea, IOem, aRangeParams)**
- Method **SetFieldAliases(cAlias, cFieldAliases)**
- Method **SetUserDataset(cAlias, cFields, bGotop, bSkipPlus1, bSkipMinus1, bCheckEof, bGetValue)**

If in the future and due user request more compatibility is requested surely it will be added to the **frReportManager()** class.

1.7 TFastReport

FastReport main report class.

The reports created by FastReport are saved in XML files with the extension FR3 and include not only the report information but important context information on where the report should be printed or shown.

Certain operations can only be made after the report is loaded. For example: since it is possible to create a new variable category, if you have not set a specific report to load, when you load it the new category will be lost and superseded with the categories of the own report. For that reason it is important to set the report filename before modifying any of its properties.

Hierarchy	TFrObject descendant
File	\source\FastReport.prg

1.7.1 TFastReport:Properties

■ Read only ■ Assignable ■ Design assignable ■ Runtime assignable

Scope	Name	Type	Initial Value
■	aDatasets	Array	{}
■	cDIIPath	String	""
■	cFileName	String	""
■	cFRLicense	String	""
■	cLanguage	Cárcater	""
■	cLastError	String	""
■	cResIco	String	""
■	cResName	String	""
■	cTitle	String	""
■	cXALicense	String	""
■	IAbortOnErrors	Logical	.T.
■	nLastError	Numeric	0
■	oEngineOptions	Object	TFrEngineOptions
■	oPreviewOptions	Object	TFrPreviewOptions
■	oPrintOptions	Object	TFrPrintOptions
■	oReportOptions	Object	TFrReportOptions

1.7.1.1 TFastReport:aDatasets

Array of TFrDataset objects connected to the report.

Scope	Read only
Type	Array
Initial Value	{}

1.7.1.2 TFastReport:cDIIPath

Frx.dll path. It must be set before the call to method Create() or on the bOnBeforeCreate event.

Scope	Assignable
Type	String
Initial Value	""

1.7.1.3 TFastReport:cFileName

Name of the FR3 or FP3 file that contains the report.

Scope	Assignable
Type	String
Initial Value	""

The set of this property provokes the deletion of the cResName property.

1.7.1.4 TFastReport:cFRLicense

FastReport license provided by FastReport Inc.

Scope	Assignable
Type	String
Initial Value	""
See also	cXALicense

This property must be assigned on every report you create in order to deactivate the evaluation mode. To avoid the work of introducing the **cFRLicense** and the **cXALicense** license, as convenient is to overload its **T** class introducing its values. For example:

```
CLASS TFastReport FROM XFastReport
  VAR cFRLicense INIT "1234-5678-9012-3456"
  VAR cXALicense INIT "XAFR-00000000-01"
END CLASS
```

1.7.1.5 TFastReport:cLanguage

Language to be used on the report designer and preview window.

Scope	Assignable
Type	String
Initial Value	""
Possible values	English, Brazil, French, German, Italian, Portuguese, Russian, Spanish, Greek

All languages literals are contained in FastReport for Xailer DLL. If your language is not available on this list, you can use the LoadLangFile method to load from an external file your own language.

1.7.1.6 TFastReport:cLastError

Last error produced.

Scope	Read only
Type	String
Initial Value	""

1.7.1.7 TFastReport:cResIco

Resource name that holds the report image (ICO).

Scope	Assignable
Type	String
Initial Value	""

1.7.1.8 TFastReport:cResName

Resource name for FR3 or FP3 report information.

Scope	Assignable
Type	String
Initial Value	""

The set of this property provokes the deletion of the cFileName property.

1.7.1.9 TFastReport:cTitle

Report title.

Scope	Assignable
Type	String
Initial Value	""

1.7.1.10 TFastReport:cXALicense

FastReport for Xailer & [x]Harbour license provided by Xailer.

Scope	Assignable
Type	String
Initial Value	""
See also	cFRLicense

This property must be assigned on every report you create in order to deactivate the evaluation mode. To avoid the work of introducing the **cXALicense** and the cFRLicense license, as convenient is to overload its **T** class introducing its values. For example:

```
CLASS TFastReport FROM XFastReport
  PROPERTY cFRLicense INIT "1234-5678-9012-3456"
  PROPERTY cXALicense INIT "XAFR-00000000-01"
END CLASS
```

1.7.1.11 TFastReport:IAbortOnErrors

If true, any error will provoke a runtime error that can be trapped with a [TRY..CATCH](#).

Scope	Assignable
Type	Logical
Initial Value	.T.

1.7.1.12 TFastReport:nLastError

Last ordinal error number produced.

Scope	Read only
Type	Logical

Initial Value	""
See also	cLastError

1.7.1.13 TFastReport:oEngineOptions

TFrEngineOptions object that holds all the properties of the report relating to the print engine.

Scope	Read only
Type	Object
Initial Value	TFrEngineOptions

1.7.1.14 TFastReport:oPreviewOptions

TFrPreviewOptions object that holds all the properties of the report relating to the preview engine.

Scope	Read only
Type	Object
Initial Value	TFrPreviewOptions

1.7.1.15 TFastReport:oPrintOptions

TFrPrintOptions object that holds all the properties of the report relating to the printing process.

Scope	Read only
Type	Object
Initial Value	TFrPreviewOptions

1.7.1.16 TFastReport:oReportOptions

TFrReportOptions object that holds all the properties of the report relating to the report itself.

Scope	Read only
Type	Object
Initial Value	TFrReportOptions

1.7.2 TFastReport:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	AddArray
■	AddCategory
■	AddDbf
■	AddFunction
■	AddVariable
■	Calc
■	CategoriesList
■	Clear
■	ClearMasterDetail
■	ClearVariables
■	Create
■	DelDataset
■	DelError
■	DeleteCategory
■	DeleteVariable
■	DesignReport
■	DoExport
■	GetErrors
■	GetProperty
■	GetVariable
■	LoadFPFile
■	LoadFromFile
■	LoadFromResource
■	LoadFromString
■	LoadLangFile
■	LoadStyleSheetFromFile
■	LoadStyleSheetFromString
■	New
■	PrepareReport
■	PreviewClose
■	Print
■	SaveToFile
■	SaveToFPFile
■	SaveToString
■	SetADOConnectStr
■	SetFormatSettings
■	SetIcon
■	SetLicense
■	SetProperty
■	SetTitle
■	SetVariable
■	ShowReport

VariablesList

1.7.2.1 TFastReport:AddArray

Adds a new dataset to the report using as data source an array.

Type	Standard
Parameters	<p><cName> Name that will be given to dataset on design time</p> <p><aData> Data array</p> <p>[<aFields>] Field names for every column on the array.</p>
Return Value	TFrArrayDataset object

1.7.2.2 TFastReport:AddCategory

Adds a new variable category to the report.

Type	After loading the report
Parameters	<p><cName> Category name</p>
Return Value	True if success

1.7.2.3 TFastReport:AddDbf

Adds a new dataset to the report using as data source one or more DBF files.

Type	Standard
Parameters	<p><cName> Name that will be given to dataset on design time</p> <p>[<aFields>] Field names for every column on the report</p>
Return Value	TFrDbfDataset object

The **aFields** parameter is an array with all the field names you want to include. There are some important features:

- You can indicate the table source prefixing the field with its alias. For example: `Clients->Code`. If you do not indicate the alias, the active area when the report is loaded is used.
- Is possible to indicate all the fields of a table using the '*' character. For example: `Clients->*`.
- Is possible to modify the name of a field for design time using the clause 'AS'. For example:

`Clients->id AS Code.`

In order to use more than one work area is necessary to set the relations between them using the xBase command `SET RELATION` or use the property TFrDataset:oDSMaster.

1.7.2.4 TFastReport:AddFunction

Adds a new function to the report. Allows that any function application is accesible to the report engine.

Type	After loading the report
Parameters	<p><code><cPrototype></code> Function prototype in PascalScript <code><cCategory></code> Category where the function should appear on the designer <code><cDescription></code> Description to be shown on the report designer</p>
Return Value	True if success

FastReport for Xailer includes in the [x]Harbour category and for compatibility reasons with the discontinued product *FastReport for [x]Harbour* some of its functions. However this is not really necessary since any function of your application can be accessed from the report generator with only this method. You only need to know how to perform the function prototyping in PascalScript.

Forexample:

```
oFR:AddFunction( "function NtoL(n: extended):String", "My category",;
    "Converts a number to a string" )
```

With this simple line you will gain access to the `NtoL()` function on your report. It's that easy. To use it in your reports can use even the wizard that provides the designer. This would be an example of its use:

```
[NtoL(<CUSTOMERS."CustNo">) ]
```

Do not worry if a runtime error occurs within your own application when you call a function defined in the report. FastReport for Xailer is able to recover itself from it and even show the error description on the report.

1.7.2.5 TFastReport:AddVariable

Adds a new variable to the report.

Type	After loading the report
Parameters	<p><code><cName></code> Variable name on the report designer <code><Value></code> Variable value</p>

Return Value	True if success
---------------------	-----------------

The variable will be created on the variables root tree unless you have create a category first with the method AddCategory.

1.7.2.6 TFastReport:Calc

Allows to calculate any PascalScript expression on the report environment. Returns the value of the expression.

Type	After loading the report
Parameters	<cExpression> Expression to evaluate
Return Value	Value

1.7.2.7 TFastReport:CategoriesList

Returns an array with all the available categories.

Type	After loading the report
Parameters	None
Return Value	altems

1.7.2.8 TFastReport:Clear

Clears the contents of the report and leave it in the same state as if he had not given any report.

Type	After loading the report
Parameters	None
Return Value	True if success

1.7.2.9 TFastReport:ClearMasterDetail

Deletes a master-detail relation.

Type	Standard
Parameters	<cMasterName> Name of master dataset <cDetailName> Name of detail dataset

Return Value	True if success
---------------------	-----------------

1.7.2.10 TFastReport:ClearVariables

Clears all the variables of the report.

Type	After loading the report
Parameters	None
Return Value	True if success

1.7.2.11 TFastReport>Create

Constructor and initializes the report engine: Loads the FRX.DLL library, sets the license codes and the language to use. The typical object creation should be as follows:

```
WITH OBJECT TFastReport () :New()
  :xProperty1 := ??
  :xProperty1 := ??
  :Create()
  :ShowReport()
END WITH
```

Type	Constructor
Parameters	None
Return Value	Self
See also	New

1.7.2.12 TFastReport:DelDataset

Deletes a dataset from the report.

Type	Standard
Parameters	<oDataset> TFrDataset object to delete
Return Value	True if success

1.7.2.13 TFastReport:DelError

Deletes last error produced.

Type	Standard
Parameters	None
Return Value	True if success

1.7.2.14 TFastReport:DeleteCategory

Deletes a report category.

Type	After loading the report
Parameters	<cName> Category name
Return Value	True if success

1.7.2.15 TFastReport:DeleteVariable

Deletes a report variable.

Type	After loading the report
Parameters	<cName> Variable name
Return Value	True if success

1.7.2.16 TFastReport:DesignReport

Runs the report designer.

Type	After loading the report
Parameters	None
Return Value	True if success

Important Note: If the report has internal master-detail operations its advisable to call the PrepareReport method first in order to achieve consistent results on preview.

1.7.2.17 TFastReport:DoExport

Exports the report with the selected filter.

Type	After loading the report
Parameters	<p><cFilter></p> <p>Filter to use.</p> <p>Possible values</p> <ul style="list-style-type: none"> PDFEXPORT HTMLEXPORT RTFEXPORT CSVEXPORT XLSEXPORT DOTMATRIXEXPORT BMPEXPORT JPGEXPORT TIFFEXPORT GIFEXPORT SIMPLETEXTEXPORT MAILEXPORT ODSEXPORT ODTEXPORT XMLEXPORT EMFEXPORT BIFFEXPORT
Return Value	True if success

For further information consult FastReport documentation.

1.7.2.18 TFastReport:GetErrors

Retrieves a array of string with all the errors produced.

Type	After loading the report
Parameters	None
Return Value	<p><aErrors></p> <p>Array of errors</p>

1.7.2.19 TFastReport:GetProperty

Retrieves any report property. Note that you can specify a complex expression as '[REPORT.ReportOptions.cAuthor](#)'.

Type	After loading the report
Parameters	<p><cObjectName></p> <p>Object name. If blank the REPORT object will be used. Possible</p>

	values BMPExport CSVExport DESIGNER DotMatrixExport GIFExport HTMLEXPORT HTMLExport JPEGExport MailExport ODSExport ODTExport PDFExport REPORT RTFExport RTFEXPORT SimpleTextExport TIFFExport XLSExport XMLEXport <cName> Variable name or expression
Return Value	Property value

1.7.2.20 TFastReport:GetVariable

Retrieves a report variable value.

Type	After loading the report
Parameters	<cName> Variable name
Return Value	Variable value

1.7.2.21 TFastReport:LoadFPFile

Loads a prepared report form file.

Type	Standard
Parameters	<cFilename> File name
Return Value	True if success

1.7.2.22 TFastReport:LoadFromFile

Loads a report from file.

Type	Standard
Parameters	<cFilename> File name
Return Value	True if success

1.7.2.23 TFastReport:LoadFromResource

Loads a report from resources.

Type	Standard
Parameters	<cResname> Resource name
Return Value	True if success

1.7.2.24 TFastReport:LoadFromString

Loads a report from a string.

Type	Standard
Parameters	<cString> String with all the report information
Return Value	True if success

1.7.2.25 TFastReport:LoadLangFile

Loads a XML file with the language to use on the report designer and preview window.

Type	Standard
Parameters	<cFilename> Filename
Return Value	True if success
See also	cLanguage

1.7.2.26 TFastReport:LoadStyleSheetFromFile

Loads a style sheet from file.

Type	After loading the report
Parameters	<cFilename> File name
Return Value	True if success

For further information consult FastReport documentation.

1.7.2.27 TFastReport:LoadStyleSheetFromString

Loads a style sheet from a string.

Type	After loading the report
Parameters	<cString> String with the information
Return Value	True if success

For further information consult FastReport documentation.

1.7.2.28 TFastReport:New

Constructor. This constructor simply creates the report object without doing any additional task.

The typical object creation should be as follows:

```
WITH OBJECT TFastReport () :New()
  :xProperty1 := ??
  :xProperty1 := ??
  :Create()
  :ShowReport()
END WITH
```

The Create() method is **not** optional.

Type	Constructor
Parameters	None
Return Value	Self
See also	Create

1.7.2.29 TFastReport:PrepareReport

Prepares the report.

Type	After loading the report
Parameters	[<IKeepLastReport>] If true previous pages from old reports will not be deleted. By default .F.
Return Value	True if success

1.7.2.30 TFastReport:PreviewClose

Closes the original FastReport preview window.

Type	After loading the report
Parameters	None
Return Value	True if success

1.7.2.31 TFastReport:Print

Prints the report.

Type	After loading the report
Parameters	None
Return Value	True if success

1.7.2.32 TFastReport:SaveToFile

Saves the report to a file.

Type	Standard
Parameters	<cFilename> File name
Return Value	True if success

1.7.2.33 TFastReport:SaveToFPFile

Saves a report with all its data to a file.

Type	Standard
Parameters	<cFilename> File name
Return Value	True if success

1.7.2.34 TFastReport:SaveToString

Saves the report to a string.

Type	Standard
Parameters	<cString> String with all the information of the report
Return Value	True if success

1.7.2.35 TFastReport:SetADOConnectStr

Sets the ADO connection string for the report.

Type	Standard
Parameters	<cADOConnect> Connection string
Return Value	True if success

1.7.2.36 TFastReport:SetFormatSettings

Sets the default value format for most locale global variables .

Type	Standard
Parameters	<nSetting> Type of setting. <xValue> Newvalue
Return Value	True if success

nSetting is described in Frh.ch:

```
#define frxCurrencyString 1
#define frxCurrencyFormat 2
#define frxNegCurrFormat 3
```

```
#define frxThousandSeparator 4
#define frxDecimalSeparator 5
#define frxCurrencyDecimals 6
#define frxDateSeparator 7
#define frxShortDateFormat 8
#define frxLongDateFormat 9
#define frxTimeSeparator 10
#define frxTimeAMString 11
#define frxTimePMString 12
#define frxShortTimeFormat 13
#define frxLongTimeFormat 14
#define frxShortMonthNames 15
#define frxLongMonthNames 16
#define frxShortDayNames 17
#define frxLongDayNames 18
#define frxTwoDigitYearCenturyWindow 20
#define frxListSeparator 21
```

For **xValue** read follow description:

The initial values of these variables are fetched from the system registry using the GetLocaleInfo function in the Win32 API. The description of each variable specifies the LOCALE_XXXX constant used to fetch the initial value.

CurrencyString - Defines the currency symbol used in floating-point to decimal conversions. The initial value is fetched from LOCALE_SCURRENCY.

CurrencyFormat - Defines the currency symbol placement and separation used in floating-point to decimal conversions. Possible values are:

```
0 = '$1'
1 = '1$'
2 = '$ 1'
3 = '1 $'
```

The initial value is fetched from LOCALE_ICURRENCY.

NegCurrFormat - Defines the currency format for used in floating-point to decimal conversions of negative numbers. Possible values are:

```
0 = '($1)' 4 = '(1$)' 8 = '-1 $' 12 = '$ -1'
1 = '-$1' 5 = '-1$' 9 = '-$ 1' 13 = '1- $'
2 = '$-1' 6 = '1-$' 10 = '1 $-' 14 = '($ 1)'
3 = '$1-' 7 = '1$-' 11 = '$ 1-' 15 = '(1 $)'
```

The initial value is fetched from LOCALE_INEGCURR.

ThousandSeparator - The character used to separate thousands in numbers with more than three digits to the left of the decimal separator. The initial value is fetched from LOCALE_STHOUSAND. A value of #0 indicates no thousand separator character should be output even if the format string specifies thousand separators.

DecimalSeparator - The character used to separate the integer part from the fractional part of a number. The initial value is fetched from LOCALE_SDECIMAL. DecimalSeparator must be a non-zero value.

CurrencyDecimals - The number of digits to the right of the decimal point in a currency amount. The initial value is fetched from LOCALE_ICURRDIGITS.

DateSeparator - The character used to separate the year, month, and day parts of a date value. The initial value is fetched from LOCATE_SDATE.

ShortDateFormat - The format string used to convert a date value to a short string suitable for editing. For a complete description of date and time format strings, refer to the documentation for the FormatDate function. The short date format should only use the date separator character and the m, mm, d, dd, yy, and yyyy format specifiers. The initial value is fetched from LOCATE_SSHORTDATE.

LongDateFormat - The format string used to convert a date value to a long string suitable for display but not for editing. For a complete description of date and time format strings, refer to the documentation for the FormatDate function. The initial value is fetched from LOCATE_SLONGDATE.

TimeSeparator - The character used to separate the hour, minute, and second parts of a time value. The initial value is fetched from LOCATE_STIME.

TimeAMString - The suffix string used for time values between 00:00 and 11:59 in 12-hour clock format. The initial value is fetched from LOCATE_S1159.

TimePMString - The suffix string used for time values between 12:00 and 23:59 in 12-hour clock format. The initial value is fetched from LOCATE_S2359.

ShortTimeFormat - The format string used to convert a time value to a short string with only hours and minutes. The default value is computed from LOCATE_ITIME and LOCATE_ITLZERO.

LongTimeFormat - The format string used to convert a time value to a long string with hours, minutes, and seconds. The default value is computed from LOCATE_ITIME and LOCATE_ITLZERO.

ShortMonthNames - Array of strings containing short month names. The mmm format specifier in a format string passed to FormatDate causes a short month name to be substituted. The default values are fetched from the LOCATE_SABBREVMONTHNAME system locale entries.

LongMonthNames - Array of strings containing long month names. The mmmm format specifier in a format string passed to FormatDate causes a long month name to be substituted. The default values are fetched from the LOCATE_SMONTHNAME system locale entries.

ShortDayNames - Array of strings containing short day names. The ddd format specifier in a format string passed to FormatDate causes a short day name to be substituted. The default values are fetched from the LOCATE_SABBREVDAYNAME system locale entries.

LongDayNames - Array of strings containing long day names. The dddd format specifier in a format string passed to FormatDate causes a long day name to be substituted. The default values are fetched from the LOCATE_SDAYNAME system locale entries.

ListSeparator - The character used to separate items in a list. The initial value is fetched from LOCATE_SLIST.

TwoDigitYearCenturyWindow - Determines what century is added to two digit years when converting string dates to numeric dates. This value is subtracted from the current year before extracting the century. This can be used to extend the lifetime of existing applications that are inextricably tied to 2 digit year data entry. The best solution to Year 2000 (Y2k) issues is not to accept 2 digit years at all - require 4 digit years in data entry to eliminate century ambiguities.

1.7.2.37 TFastReport:SetIcon

Sets the resource icon from resources.

Type	Standard
Parameters	<cResname> Resource name. Must be BMP type
Return Value	True if success

1.7.2.38 TFastReport:SetLicense

Sets the license to deactivate the evaluation mode.

Type	Standard
Parameters	<cFRLicense> FastReport license <cXALicense> FastReport for Xailer license
Return Value	True if success
See also	cFRLicense, cXALicense

1.7.2.39 TFastReport:SetProperty

Sets a report property. Note that you can use complex expression like '`REPORT.ReportOptions.cAuthor = 'Peter'`'.

Type	After loading the report
Parameters	<cObjectName> Object name. If blank the <code>REPORT</code> object will be used. Possible values <code>BMPElexport</code> <code>CSVExport</code> <code>DESIGNER</code> <code>DotMatrixExport</code> <code>GIFExport</code> <code>HTMLEXPORT</code> <code>HTMLExport</code> <code>JPEGExport</code> <code>MailExport</code> <code>ODSExport</code> <code>ODTExport</code> <code>PDFExport</code> <code>REPORT</code>

	RTFExport RTFEXPORT SimpleTextExport TIFFExport XLSExport XMLExport <cName> Variable name or expression <xValue> Newvalue
Return Value	NIL

1.7.2.40 TFastReport:SetTitle

Sets report title.

Type	Standard
Parameters	<cTitle> Report title
Return Value	True if success

1.7.2.41 TFastReport:SetVariable

Sets the value of a report variable.

Type	After loading the report
Parameters	<cName> Variable name <xValue> Newvalue
Return Value	True if success

1.7.2.42 TFastReport>ShowReport

Shows report preview.

Type	After loading the report
Parameters	[<IKeepLastReport>] If true previous preview pages from old reports will not be deleted. By default .F.
Return Value	True if success

1.7.2.43 TFastReport:VariablesList

Returns an array with all the variables of a defined category.

Type	After loading the report
Parameters	<cCategory> Category name
Return Value	altems

1.7.3 TFastReport:Events

Name
bOnAfterLoad
bOnAfterPrint
bOnAfterPrintReport
bOnBeforeConnect
bOnBeforeCreate
bOnBeforeLoad
bOnBeforePrint
bOnBeginDoc
bOnBtnGenPdf
bOnBtnPrint
bOnBtnSendMail
bOnClickObject
bOnEndDoc
bOnEndPreview
bOnGetValue
bOnMouseOverObject
bOnPageChanged
bOnPreview
bOnPrintPage
bOnPrintReport
bOnProgress
bOnProgressStart
bOnProgressStop

1.7.3.1 TFastReport:bOnAfterLoad

Event that fires when the report is loaded. This is a good place to change the value of any property of the report.

Parameters	<oSender>: Object that triggers the event.
Return Value	NIL

1.7.3.2 TFastReport:bOnAfterPrint

Event that fires every time you print any report object.

Parameters	<oSender>: TFastReport object that triggers the event <cObject> Object name
Return Value	NIL

1.7.3.3 TFastReport:bOnAfterPrintReport

Event that is fired after the report is complete.

Parameters	<oSender>: Object that triggers the event.
Return Value	NIL

1.7.3.4 TFastReport:bOnBeforeConnect

Event that fires whenever a connection is made with a ADO connector.

Parameters	<oSender>: TFastReport object that triggers the event <cObject> ADO object name
Return Value	NIL

1.7.3.5 TFastReport:bOnBeforeCreate

Event that fires when the control is created and before the DLL is loaded.

Parameters	<oSender>: TFastReport object that triggers the event
Return Value	NIL

1.7.3.6 TFastReport:bOnBeforeLoad

Event that fires before the report is loaded but after all data and its relations are loaded.

Parameters	<oSender>: Object that triggers the event.
Return Value	NIL

1.7.3.7 TFastReport:bOnBeforePrint

Event that fires before any object gets printed.

Parameters	<oSender>: Object that triggers the event. <cObject> Object name
Return Value	NIL

1.7.3.8 TFastReport:bOnBeginDoc

Event that triggers when you start creating the report document.

Parameters	<oSender>: Object that triggers the event.
Return Value	NIL

1.7.3.9 TFastReport:bOnBtnGenPdf

Event that triggers when you push the PDF button on the preview window.

Parameters	<oSender>: Object that triggers the event.
Return Value	If a value different from NIL is returned the default button operation will be canceled.

1.7.3.10 TFastReport:bOnBtnPrint

Event that triggers when you push the 'Print' button on the preview window.

Parameters	<oSender>:
-------------------	------------

	Object that triggers the event.
Return Value	If a value different from NIL is returned the default button operation will be canceled.

1.7.3.11 TFastReport:bOnBtnSendMail

Event that triggers when you push the 'Send mail' button on the preview window.

Parameters	<oSender>: Object that triggers the event.
Return Value	If a value different from NIL is returned the default button operation will be canceled.

1.7.3.12 TFastReport:bOnClickObject

Event that is fired whenever you click on any object in the report.

Parameters	<oSender>: TFastReport object that triggers the event <cObject> Object name
Return Value	NIL

1.7.3.13 TFastReport:bOnEndDoc

Event that fires when you finish creating the report document.

Parameters	<oSender>: Object that triggers the event.
Return Value	NIL

1.7.3.14 TFastReport:bOnEndPreview

Event that fires when you close the report preview.

Parameters	<oSender>: Object that triggers the event.
Return Value	NIL

1.7.3.15 TFastReport:bOnGetValue

Event that is fired before evaluating a variable. If this event is overloaded you can change the value of it with the return value.

Parameters	<oSender>: TFastReport object that triggers the event <cVar> Variable name
Return Value	<xValue> newvalue

1.7.3.16 TFastReport:bOnMouseOverObject

Event that is fired whenever the mouse is positioned over any object of the report.

Parameters	<oSender>: TFastReport that triggers the event <cObject> Object name
Return Value	NIL

1.7.3.17 TFastReport:bOnPageChanged

Event that fires every time the page is changed on the preview window.

Parameters	<oSender>: TFastReport object that triggers the event <nPage> Page number
Return Value	NIL

1.7.3.18 TFastReport:bOnPreview

Event that triggers at the beginning of the preview of the report.

Parameters	<oSender>: Object that triggers the event.
Return Value	NIL

1.7.3.19 TFastReport:bOnPrintPage

Event that fires before each page of the report.

Parameters	<oSender>: TFastReport object that triggers the event <nCopyPage> Page copy number
Return Value	NIL

1.7.3.20 TFastReport:bOnPrintReport

Event that fires when the report begins printing.

Parameters	<oSender>: Object that triggers the event.
Return Value	NIL

1.7.3.21 TFastReport:bOnProgress

Event that is fired to mark the progress of the report.

Parameters	<oSender>: TFastReport object that triggers the event <nProgressType> 0 Building 1 Exporting 2 Printing <nProgress> Page number in process
Return Value	NIL

1.7.3.22 TFastReport:bOnProgressStart

Event that fires when the report starts a process.

Parameters	<oSender>: TFastReport object that triggers the event <nProgressType> 0 Building 1 Exporting 2 Printing <nProgress> Page number in process
-------------------	--

Return Value	NIL
---------------------	-----

1.7.3.23 TFastReport:bOnProgressStop

Event that fires when interrupting the process of creating the report.

Parameters	<oSender> : TFastReport object that triggers the event <nProgressType> 0 Building 1 Exporting 2 Printing <nProgress> Page number in process
Return Value	NIL

1.8 TFrDataset

Class to manage all the TFastReport data connectors.

This class is responsible for making the connection between FastReport reports and the data handled by our application. It supports two types of data sources and for each of them there is a specialized class that inherits from **TFrDataset**, which are:

- TFrDbfDataset
- TFrArrayDataset

In no case should directly instantiate objects of this class.

Hierarchy	TFrObject descendant
File	\source\FrDataset.prg

1.8.1 TFrDataset:Properties

■ Read only ■ Assignable ■ Design assignable ■ Runtime assignable

Scope	Name	Type	Initial Value
■	aRelationFields	Array	{}
■	cName	String	""
■	nLoaded	Numeric	0
■	nMaxRecsOnDesign	Numeric	100
■	oDsMaster	Object	NIL
■	oReport	Object	NIL

1.8.1.1 TFrDataset:aRelationFields

String array with all the master-detail relations to be used in conjuntio with the oDsMaster property.

Sets the fields to use for the master-detail relation. Each field must occupy a single array element. In case the field names do not match in both datasets you will have to use the following expression to indicate it:

```
'cDetailField=cMasterField'
```

In order to achieve a special ordering on the detail dataset you can include any extra field (last array element) on the detail field side separated by ':'. For example:

```
'cDetailField1;cDetailField2=cMasterField1'
```

Scope	Standard
Type	Array
Initial Value	{}

1.8.1.2 TFrDataset:cName

Dataset name on the report designer. This property must be set with any value In case is not assigned a default value will be given.

This property must be sent before the report is loaded.

Scope	Design assignable
Type	String
Initial Value	""

1.8.1.3 TFrDataset:nLoaded

Indicates the data load state of the dataset.

Scope	Read only
Type	Numeric
Initial Value	0
Possible values	0 Not loaded 1 Loaded for design 2 Complete load

1.8.1.4 TFrDataset:nMaxRecsOnDesign

Sets the maximum number of records that will be downloaded when using the report designer. This property allows you to limit the number of records to be loaded for design processes to not delay carrying large amount of data. A value of zero means to download the complete set of records.

Scope	Design assignable
Type	Numeric
Initial Value	100

1.8.1.5 TFrDataset:oDsMaster

TFrDataset inherited object that sets the 'Master' dataset on a master-detail relation.

Scope	Standard
Type	TFrDataset object
Initial Value	NIL
See also	aRelationFields

1.8.1.6 TFrDataset:oReport

TFastReport owner of the dataset. Must be assigned before calling its **Create** method.

Scope	Design assignable
Type	TFastReport object
Initial Value	NIL

1.8.2 TFrDataset:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	IsActive
■	IsLoaded
■	Refresh
■	SetMaster

1.8.2.1 TFrDataset:IsActive

Returns true if the dataset is active and assigned to its oReport object.

Type	Standard
Parameters	None
Return Value	True if assigned

1.8.2.2 TFrDataset:IsLoaded

Returns true if the dataset is already loaded.

Type	Standard
Parameters	None
Return Value	True if loaded

1.8.2.3 TFrDataset:Refresh

It causes a reload of the data.

Type	Standard
Parameters	None
Return Value	True if success

1.8.2.4 TFrDataset:SetMaster

Sets a master-detail relation.

Type	Standard
Parameters	<p><oMasterDataset> Master TFrdataset object <aFields> Fields on both datasets that will be used to set the relation. Each field must occupy an element in the array. When both fields name are different you can use the following expression to indicate it: <code>'cDetailField=cMasterField'</code></p>
Return Value	NIL

1.8.3 TFrDataset:Events

Name

bOnAfterLoad
bOnClose
bOnFirst
bOnNext
bOnOpen
bOnPrior

1.8.3.1 TFrDataset:bOnAfterLoad

Event that fires when the report is loaded. This is a good place to change the value of any property of the report.

Parameters	<oSender>: Object that triggers the event.
Return Value	NIL

1.8.3.2 TFrDataset:bOnClose

Event that fires when you close the dataset.

Parameters	<oSender>: Object that triggers the event.
Return Value	NIL

1.8.3.3 TFrDataset:bOnFirst

Event that triggers when the dataset is positioned at its first record.

Parameters	<oSender>: Object that triggers the event.
Return Value	NIL

1.8.3.4 TFrDataset:bOnNext

Event that triggers when positioning the dataset on next record.

Parameters	<oSender>: Object that triggers the event.
-------------------	---

Return Value	NIL
---------------------	-----

1.8.3.5 TFrDataset:bOnOpen

Event that triggers when you open the dataset.

Parameters	<oSender>: Object that triggers the event.
Return Value	NIL

1.8.3.6 TFrDataset:bOnPrior

Event that triggers when positioning the dataset on the previous record.

Parameters	<oSender>: Object that triggers the event.
Return Value	NIL

1.9 TFrDbfDataset

Class specialized on connecting DBF files with FastReport.

Hierarchy	TFrDataset descendant
File	\source\FrDataset.prg

1.9.1 TFrDbfDataset:Properties

■ Read only ■ Assignable ■ Design assignable ■ Runtime assignable

Scope	Name	Type	Initial Value
■	ILoadOnDemand	Logical	.F.
■	aFields	Array	{}

1.9.1.1 TFrDbfDataset:aFields

The **aFields** is an array with all the field names you want to include. There are some important features:

- You can indicate the table source prefixing the field with its alias. For example: [Clients->Code](#). If you do not indicate the alias, the active area when the report is loaded is

used.

- Is possible to indicate all the fields of a table using the '*' character. For example:
`Clients->*`.
- Is possible to modify the name of a field for design time using the clause 'AS'. For example:
`Clients->id AS Code`.

In order to use more than one work area is necessary to set the relations between them using the xBase command `SET RELATION`.

Scope	Design assignable
Type	Array
Initial Value	{}

1.9.1.2 TFrDbfDataset:ILoadOnDemand

If true the report data will be loaded on demand. This specially useful on very large tables because Fast-Report normally can begin printing without having all the report data. In most cases should not be necessary its use.

Important note: You must not change the DBF table state during the printing process. Avoid to use tables that are also displayed within a browse at the same time.

Scope	Design assignable
Type	Logical
Initial Value	.F.

1.10 TFrArrayDataset

Class specialized on connecting arrays with FastReport..

Hierarchy	TFrDataset descendant
File	\source\FrDataset.prg

1.10.1 TFrArrayDataset:Properties

Read only Assignable Design assignable Runtime assignable

Scope	Name	Type	Initial Value
	aData	Array	{}
	aFields	Array	{}

1.10.1.1 TFrArrayDataset:aData

Array containing the data to be used by report.

Scope	Design assignable
Type	Array
Initial Value	{}

1.10.1.2 TFrArrayDataset:aFields

Field names for every array column

Scope	Design assignable
Type	Array
Initial Value	{}

1.11 TFrEngineOptions

Class for handling the FastReport print engine.

All the **TFrEngine???** classes indeed they inherit from a base class with the name **TFrOptions**, but is not documented since none of its members can be used directly.

Hierarchy	TFrObject descendant
File	\source\FrOptions.prg

1.11.1 TFrEngineOptions:Properties

■ Read only ■ Assignable ■ Design assignable ■ Runtime assignable

Scope	Name	Type	Initial Value
■	cTempDir	String	""
■	lConvertNulls	Logical	.T.
■	lDoublePass	Logical	.F.
■	lIgnoreDevByZero	Logical	.F.
■	lPrintIfEmpty	Logical	.T.
■	lSilentMode	Logical	.F.
■	lUseFileCache	Logical	.F.
■	nMaxMemSize	Numeric	10
■	nNewSilentMode	Numeric	simMessageBoxes (0)

1.11.1.1 TFrEngineOptions:cTempDir

Specifies a path to the directory for storing temporary files.

Scope	Assignable
Type	String
Initial Value	""

1.11.1.2 TFrEngineOptions:IConvertNulls

Converts the “Null” value of the DB field into “0,” “False,” or empty string, depending on the field type.

Scope	Assignable
Type	Logical
Initial Value	.T.

1.11.1.3 TFrEngineOptions:IDoublePass

Makes a report a two-pass one.

Scope	Assignable
Type	Logical
Initial Value	.F.

1.11.1.4 TFrEngineOptions:IgnoreDevByZero

Ignores division by zero errors.

Scope	Assignable
Type	Logical
Initial Value	.F.

1.11.1.5 TFrEngineOptions:IPrintIfEmpty

Defines, whether it is necessary to print a blank report (one which containing no data lines).

Scope	Assignable
Type	Logical

Initial Value	.T.
----------------------	-----

1.11.1.6 TFrEngineOptions:ISilentMode

“Silent” mode. Thus all messages about errors can only be retrieved with the GetErrors method.

Scope	Assignable
Type	Logical
Initial Value	.F.

1.11.1.7 TFrEngineOptions:IUseFileCache

Defines, whether it is necessary to use report pages caching into the file (see the “nMaxMemSize” property).

Scope	Assignable
Type	Logical
Initial Value	.F.

1.11.1.8 TFrEngineOptions:nMaxMemSize

The maximum size of memory in Mbytes, allocated to the report pages’ cache. It becomes useful in cases when the “IUseFileCache” property is equal to .T.. If a report begins to occupy more memory during construction, caching of the constructed report pages into a temporary file is performed. This property is inexact and allows only approximate determination of the memory limit.

Scope	Assignable
Type	Numeric
Initial Value	10

1.11.1.9 TFrEngineOptions:nNewSilentMode

Property to set ISilentMode with a better precision.

Scope	Assignable
Type	Numeric
Initial Value	<code>simMessageBoxes</code>
Possible values	<code>simMessageBoxes (0)</code> Shows a dialog for every error occurred

	<p><code>simSilent</code> (1): Same behaviour that <code>ISilentMode</code> set to true <code>simReTrhow</code> (2): Forces a runtime error</p>
--	---

1.12 TFrPreviewOptions

Class for handling the FastReport preview engine.

All the **TFrEngine???** classes indeed they inherit from a base class with the name **TFrOptions**, but is not documented since none of its members can be used directly.

Hierarchy File	TFrObject descendant \source\FrOptions.prg
---------------------------------	---

1.12.1 TFrPreviewOptions:Properties

■ Read only ■ Assignable ■ Design assignable ■ Runtime assignable

Scope	Name	Type	Initial Value
■	<code>IAllowEdit</code>	Logical	.F.
■	<code>IDoubleBuffered</code>	Logical	.T.
■	<code>IMaximized</code>	Logical	.T.
■	<code>IModal</code>	Logical	.T.
■	<code>IOutlineExpand</code>	Logical	.T.
■	<code>IOutlineVisible</code>	Logical	.F.
■	<code>IPictureCacheInFile</code>	Logical	.F.
■	<code>IShowCaptions</code>	Logical	.F.
■	<code>nButtons</code>	Numeric	4095
■	<code>nOutLineWidth</code>	Numeric	180
■	<code>nZoom</code>	Numeric	1.00

1.12.1.1 TFrPreviewOptions:IAllowEdit

Enables or disables a finished report editing. Always false in DEMO versions.

Scope	Assignable
Type	Logical
Initial Value	.F.

1.12.1.2 TFrPreviewOptions:IDoubleBuffered

A double-buffer mode for the preview window. If enabled (by default), the preview window will not flicker during repainting, but the process speed would be reduced.

Scope	Assignable
Type	Logical
Initial Value	.T.

1.12.1.3 TFrPreviewOptions:IMaximized

Defines whether the preview window is maximized.

Scope	Assignable
Type	Logical
Initial Value	.T.

1.12.1.4 TFrPreviewOptions:IModal

Defines whether the preview window is modal.

Scope	Assignable
Type	Logical
Initial Value	.T.

1.12.1.5 TFrPreviewOptions:IOutlineExpand

Defines whether the report outline is expanded or not.

Scope	Assignable
Type	Logical
Initial Value	.T.

1.12.1.6 TFrPreviewOptions:IOutlineVisible

Defines whether the panel with the report outline is visible.

Scope	Assignable
Type	Logical

Initial Value	.F.
----------------------	-----

1.12.1.7 TFrPreviewOptions:IPictureCacheInFile

Defines whether images should be saved on cache.

Scope	Assignable
Type	Logical
Initial Value	.F.

1.12.1.8 TFrPreviewOptions:IShowCaptions

Defines whether it is necessary to display button captions. When enabling this property, you should limit the number of the displayed buttons in the nButtons property, since all the buttons would not find room on the screen

Scope	Assignable
Type	Logical
Initial Value	.F.

1.12.1.9 TFrPreviewOptions:nButtons

A set of buttons, which will be available in the preview window.

Scope	Assignable
Type	Numeric
Initial Value	4095
Possible values	Any nOr() combination of the following values: pbPrint 1 pbLoad 2 pbSave 4 pbExport 8 pbZoom 16 pbFind 32 pbOutline 64 pbPageSetup 128 pbTools 256 pbEdit 512 pbNavigator 1024 pbExportQuick 2048 pbNoClose 4096 pbNoFullScreen 8192 pbNoEmail 16384

1.12.1.10 TFrPreviewOptions:nOutlineWidth

Defines width of the panel with the report outline.

Scope	Assignable
Type	Numeric
Initial Value	180

1.12.1.11 TFrPreviewOptions:nZoom

The default zooming value.

Scope	Assignable
Type	Numeric
Initial Value	1.00

1.12.2 TFrPreviewOptions:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	SetBounds

1.12.2.1 TFrPreviewOptions:SetBounds

Sets the preview window coordinates.

Type	Standard
Parameters	<p><nLeft> Left coordinate <nTop> Top coordinate <nWidth> Window width in pixels <nHeight> Window height in pixels</p>
Return Value	NIL

1.13 TFrPrintOptions

Class for handling the FastReport printing.

All the **TFrEngine???** classes indeed they inherit from a base class with the name **TFrOptions**, but is not documented since none of its members can be used directly.

Hierarchy	TFrObject descendant
File	\source\FrOptions.prg

1.13.1 TFrPrintOptions:Properties

■ Read only ■ Assignable ■ Design assignable ■ Runtime assignable

Scope	Name	Type	Initial Value
■	cPageNumbers	String	""
■	cPrinter	String	"Default"
■	ICollate	Logical	.T.
■	IReverse	Logical	.F.
■	IShowDialog	Logical	.T.
■	nCopies	Numeric	1
■	nPrintPages	Numeric	ppAll

1.13.1.1 TFrPrintOptions:cPageNumbers

Page numbers, which are to be printed. For example, "1,3,5-12,17-".

Scope	Assignable
Type	String
Initial Value	""

1.13.1.2 TFrPrintOptions:cPrinter

Printer name.

Scope	Assignable
Type	String
Initial Value	"Default"

1.13.1.3 TFrPrintOptions:ICollate

Whether to collate the copies.

Scope	Assignable
Type	Logical
Initial Value	.T.

1.13.1.4 TFrPrintOptions:IReverse

Whether to reverse page printing.

Scope	Assignable
Type	Logical
Initial Value	.F.

1.13.1.5 TFrPrintOptions:IShowDialog

Whether to display a print dialogue.

Scope	Assignable
Type	Logical
Initial Value	.T.

1.13.1.6 TFrPrintOptions:nCopies

Number of copies.

Scope	Assignable
Type	Numeric
Initial Value	1

1.13.1.7 TFrPrintOptions:nPrintPages

Defines the pages to be printed.

Scope	Assignable
Type	Numeric
Initial Value	ppAll

Possible values	ppAll ppOdd ppEven	0 1 2
------------------------	--------------------------	-------------

1.13.2 TFrPrintOptions:Methods

■ Constructor ■ Standard ■ Only after Create()

Type	Name
■	ClearOptions

1.13.2.1 TFrPrintOptions:ClearOptions

Resets all the properties to their default value.

Type	Standard
Parameters	None
Return Value	NIL

1.14 TFrReportOptions

Class for handling the report itself.

All the **TFrEngine???** classes indeed they inherit from a base class with the name **TFrOptions**, but is not documented since none of its members can be used directly.

Hierarchy	TFrObject descendant
File	\source\FrOptions.prg

1.14.1 TFrReportOptions:Properties

■ Read only ■ Assignable ■ Design assignable ■ Runtime assignable

Scope	Name	Type	Initial Value
■	aDescription	Array	{}
■	cAuthor	String	""
■	cInitString	String	""
■	cPasword	String	""
■	lCompressed	Logical	.F.

1.14.1.1 TFrReportOptions:aDescription

String array with report description.

Scope	Assignable
Type	Array
Initial Value	{}

1.14.1.2 TFrReportOptions:cAuthor

Report author.

Scope	Assignable
Type	Cadena
Initial Value	""

1.14.1.3 TFrReportOptions:cInitString

Escape sequence for dot matrix printers.

Scope	Assignable
Type	Cadena
Initial Value	""

1.14.1.4 TFrReportOptions:cPassword

Password for encrypted reports.

Scope	Assignable
Type	Cadena
Initial Value	""

1.14.1.5 TFrReportOptions:lCompressed

Whether the report is compressed.

Scope	Assignable
Type	Logical
Initial Value	.F.

Índice

- : -

:SetMaster 50

- A -

aData 54
 aDatasets 20
 AddArray 26
 AddCategory 26
 AddDbf 26
 AddFunction 27
 AddVariable 27
 aDescription 64
 aFields 52, 54
 aRelationFields 48

- B -

bOnAfterLoad 41, 51
 bOnAfterPrint 42
 bOnAfterPrintReport 42
 bOnBeforeConnect 42
 bOnBeforeCreate 42
 bOnBeforeLoad 43
 bOnBeforePrint 43
 bOnBeginDoc 43
 bOnBtnPrint 43
 bOnClickObject 44
 bOnClose 51
 bOnEndDoc 44
 bOnEndPreview 44
 bOnFirst 51
 bOnGetValue 45
 bOnMouseOverObject 45
 bOnNext 51
 bOnOpen 52
 bOnPageChanged 45
 bOnPreview 45
 bOnPrintPage 46
 bOnPrintReport 46

bOnPrior 52
 bOnProgress 46
 bOnProgressStart 46
 bOnProgressStop 47

- C -

Calc 28
 CategoriesList 28
 cAuthor 64
 cDIIPath 21
 cFileName 21
 cFRLicense 21
 cInitString 64
 cLanguage 22
 cLastError 22
 Clear 28
 ClearMasterDetail 28
 ClearOptions 63
 ClearVariables 29
 cName 48
 cPageNumbers 61
 cPassword 64
 cPrinter 61
 Create 29
 cResIco 22
 cResName 22
 cTempDir 55
 cTitle 23
 cXALicense 23

- D -

DelDataset 29
 DelError 30
 DeleteCategory 30
 DeleteVariable 30
 DoExport 31

- F -

FastReport 9
- G -
 GetErrors 31

GetProperty 31
GetVariable 32

- I -

IsActive 50
IsLoaded 50

- L -

IAbortOnErrors 23
IAllowEdit 57
ICollate 62
ICompressed 64
IConvertNulls 55
IDoubleBuffered 58
IDoublePass 55
License 13
IIgnoreDevByZero 55
Linking 12
ILoadOnDemand 53
IMaximized 58
IModal 58
LoadFPFile 32
LoadFromFile 33
LoadFromResource 33
LoadFromString 33
LoadLangFile 33
LoadStyleSheetFromFile 34
LoadStyleSheetFromString 34
IOutlineExpand 58
IOutlineVisible 58
IPictureCacheInFile 59
IPrintIfEmpty 55
IReverse 62
IShowDialog 62
ISilentMode 56
IUseFileCache 56

- N -

nButtons 59
nCopies 62
New 34
nLastError 23

nLoaded 48
nMaxMemSize 56
nMaxRecsOnDesign 49
nNewSilentMode 56
nOutlineWidth 60
nPrintPages 62
nZoom 60

- O -

oDsMaster 49
oEngineOptions 24
OnBtnGenPdf 43
OnBtnSendMail 44
OnPreview 41
oPreviewOptions 24
oPrintOptions 24
oReport 49
oReportOptions 24

- P -

PrepareReport 35
PreviewClose 35
Print 35

- R -

Refresh 50
Registration 12

- S -

SaveToFile 35
SaveToFPFile 36
SaveToString 36
Sergey Spirin compatibility 19
SetADOConnectStr 36
SetBounds 60
SetFormatSettings 36
SetIcon 39
SetLicense 39
 SetProperty 39
SetVariable 40
ShowReport 40

- T -

T Classes & X Classes 18
TFromArrayDataset:aData 54
TFastReport 20
TFastReport:aDatasets 20
TFastReport:AddCategory 26
TFastReport:AddDbf 26
TFastReport:AddFunction 27
TFastReport:AddVariable 27
TFastReport:bOnAfterPrint 42
TFastReport:bOnAfterPrintReport 42
TFastReport:bOnBeforeConnect 42
TFastReport:bOnBeforeCreate 42
TFastReport:bOnBeforeLoad 43
TFastReport:bOnBeforePrint 43
TFastReport:bOnBeginDoc 43
TFastReport:bOnBtnGenPdf 43
TFastReport:bOnBtnPrint 43
TFastReport:bOnBtnSendMail 44
TFastReport:bOnClickObject 44
TFastReport:bOnEndDoc 44
TFastReport:bOnGetValue 45
TFastReport:bOnMouseOverObject 45
TFastReport:bOnPageChanged 45
TFastReport:bOnPreview 45
TFastReport:bOnPrintPage 46
TFastReport:bOnPrintReport 46
TFastReport:bOnProgress 46
TFastReport:bOnProgressStart 46
TFastReport:bOnProgressStop 47
TFastReport:Calc 28
TFastReport:CategoriesList 28
TFastReport:cDIIPath 21
TFastReport:cFileName 21
TFastReport:cFRLicense 21
TFastReport:cLanguage 22
TFastReport:cLastError 22
TFastReport:Clear 28
TFastReport:ClearMasterDetail 28
TFastReport:ClearVariables 29
TFastReport>Create 29
TFastReport:cResIco 22
TFastReport:cResName 22
TFastReport:cTitle 23
TFastReport:cXALicense 23
TFastReport:DelDataset 29
TFastReport:DelError 30
TFastReport>DeleteCategory 30
TFastReport>DeleteVariable 30
TFastReport:DoExport 31
TFastReport:Events 41
TFastReport:GetErrors 31
TFastReport:GetProperty 31
TFastReport:GetVariable 32
TFastReport:IAbortOnErrors 23
TFastReport:LoadFPFile 32
TFastReport:LoadFromFile 33
TFastReport:LoadFromResource 33
TFastReport:LoadFromString 33
TFastReport:LoadLangFile 33
TFastReport:LoadStyleSheetFromFile 34
TFastReport:LoadStyleSheetFromString 34
TFastReport:Methods 25
TFastReport>New 34
TFastReport:nLastError 23
TFastReport:oEngineOptions 24
TFastReport:OnEndPreview 44
TFastReport:oPreviewOptions 24
TFastReport:oPrintOptions 24
TFastReport:oReportOptions 24
TFastReport:PrepareReport 35
TFastReport:PreviewClose 35
TFastReport:Print 35
TFastReport:Properties 20
TFastReport:SaveToFile 35
TFastReport:SaveToFPFile 36
TFastReport:SaveToString 36
TFastReport:SetADOConnectStr 36
TFastReport:SetFormatSettings 36
TFastReport:SetIcon 39
TFastReport:SetLicense 39
TFastReport:SetProperty 39
TFastReport:SetVariable 40
TFastReport>ShowReport 40
TFastReport:VariablesList 41
TFastReportAddArray 26
TFastReport:bOnAfterLoad 41
TFrArrayDataset 53

TFrArrayDataset:aFields 54
TFrArrayDataset:Properties 53
TFrDataset 47
TFrDataset:aRelationFields 48
TFrDataset:bOnAfterLoad 51
TFrDataset:bOnClose 51
TFrDataset:bOnFirst 51
TFrDataset:bOnNext 51
TFrDataset:bOnOpen 52
TFrDataset:bOnPrior 52
TFrDataset:cName 48
TFrDataset:Events 51
TFrDataset:IsActive 50
TFrDataset:IsLoaded 50
TFrDataset:Methods 49
TFrDataset:nLoaded 48
TFrDataset:nMaxRecsOnDesign 49
TFrDataset:oDsMaster 49
TFrDataset:oReport 49
TFrDataset:Properties 47
TFrDataset:Refresh 50
TFrDataset:SetMaster 50
TFrDbfDataset 52
TFrDbfDataset:aFields 52
TFrDbfDataset:ILoadOnDemand 53
TFrDbfDataset:Properties 52
TFrEngineOptions 54
TFrEngineOptions:cTempDir 55
TFrEngineOptions:IDoublePass 55
TFrEngineOptions:IgnoreDevByZero 55
TFrEngineOptions:IPrintIfEmpty 55
TFrEngineOptions:ISilentMode 56
TFrEngineOptions:IUseFileCache 56
TFrEngineOptions:nMaxMemSize 56
TFrEngineOptions:nNewSilentMode 56
TFrEngineOptions:Properties 54
TFrEngineOptionsn:IConvertNulls 55
TFrPreviewOptions 57
TFrPreviewOptions:IAccordEdit 57
TFrPreviewOptions:IDoubleBuffered 58
TFrPreviewOptions:IMaximized 58
TFrPreviewOptions:IModal 58
TFrPreviewOptions:IOutlineExpand 58
TFrPreviewOptions:IOutlineVisible 58
TFrPreviewOptions:IPictureCacheInFile 59
TFrPreviewOptions:Methods 60
TFrPreviewOptions:nButtons 59
TFrPreviewOptions:nOutlineWidth 60
TFrPreviewOptions:nZoom 60
TFrPreviewOptions:Properties 57
TFrPreviewOptions:SetBounds 60
TFrPrintOptions 61
TFrPrintOptions:ClearOptions 63
TFrPrintOptions:cPageNumbers 61
TFrPrintOptions:cPrinter 61
TFrPrintOptions:ICollate 62
TFrPrintOptions:IRReverse 62
TFrPrintOptions:IShowDialog 62
TFrPrintOptions:Methods 63
TFrPrintOptions:nCopies 62
TFrPrintOptions:nPrintPages 62
TFrPrintOptions:Properties 61
TFrReportOptions 63
TFrReportOptions:aDescription 64
TFrReportOptions:cAuthor 64
TFrReportOptions:cInitString 64
TFrReportOptions:cPassword 64
TFrReportOptions:ICompressed 64
TFrReportOptions:Properties 63

- V -

VariablesList 41

- X -

Xailer vs Standard version 18