



**HAL**  
open science

# MODE-PP TeX: A GRM/GDMO Pretty Printing Library based on MODE-FE for the Generation of TeX Documents: Release 1.0 Reference Manual

Olivier Festor

► **To cite this version:**

Olivier Festor. MODE-PP TeX: A GRM/GDMO Pretty Printing Library based on MODE-FE for the Generation of TeX Documents: Release 1.0 Reference Manual. [Technical Report] RT-0192, INRIA. 1996, pp.17. inria-00069979

**HAL Id: inria-00069979**

**<https://inria.hal.science/inria-00069979>**

Submitted on 19 May 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

***MODE-PP TeX: A GRM/GDMO Pretty Printing Library  
based on MODE-FE for the Generation of TeX  
Documents.  
-Release 1.0-  
Reference Manual***

Olivier Festor

**N° 0192**

Avril 1996

PROGRAMME 1



*Rapport  
technique*



# MODE-PP TeX: A GRM/GDMO Pretty Printing Library based on MODE-FE for the Generation of TeX Documents. -Release 1.0- Reference Manual

Olivier Festor

Programme 1 — Architectures parallèles, bases de données, réseaux et systèmes distribués  
Projet RESEDAS

Rapport technique n° 0192 — Avril 1996 — 17 pages

**Abstract:** MODE-PP TeX (MODE TeX Pretty-Printer) is a module which allows formatting and pretty printing of GRM (General Relationship Model) and GDMO (Guidelines for the Definition of Managed Objects) specifications. The tool allows the generation of one or more TeX files which contain formatted specifications according to different document style constrains.

MODE-PP TeX is part of the MODE (Managed Object Development Environment) prototype which is developed for integrating the use of Formal Description Techniques in an environment for building Management applications. MODE-PP TeX is built on top of MODE-FE and is provided as a library. This document is the reference manual for release 1.0 of the MODE-PP TeX library and its API.

**Key-words:** GDMO, GRM, MODE, TeX, Reference Manual

*(Résumé : tsvp)*

**MODE-PP TeX: Un formateur GRM/GDMO basé sur MODE-FE  
pour la génération de documents TeX  
-Version 1.0-  
Manuel de référence**

**Résumé :**

MODE-PP TeX (MODE TeX Pretty-Printer) est un module de formatage de spécifications GRM (General Relationship Model) et GDMO (Guidelines for the Definition of Managed Objects). Il permet de générer un ou plusieurs fichiers TeX comportant des spécifications formatées suivant plusieurs canevas.

MODE-PP TeX est une application de l'environnement MODE développé pour intégrer l'utilisation des méthodes de description formelle dans le développement d'applications de gestion de réseaux. MODE-PP-TeX est construit sur la librairie MODE-FE. Il est disponible sous forme d'une librairie. Ce document forme le manuel de référence de la version 1.0 de la librairie MODE-PP et de son interface de programmation.

**Mots-clé :** GDMO, GRM, MODE, TeX, Manuel de Référence

# Chapter 1

## Introduction

This document forms the reference manual of the **MODE-PP TeX** (release 1.0) part of the **MODE** environment. **MODE** stands for **Managed Object Development Environment** and is a research prototype under development within the RESEDAS research group of the INRIA Lorraine and CRIN/CNRS. The **MODE** environment is designed for supporting various developments based on latest OSI standards for Management Information Modelling combined with Formal Description Techniques.

The **MODE-PP** (**MODE Pretty-Printer**) is built on top of the **MODE-FE Library** [Festor 96]. **MODE-PP** provides formatting and pretty-printing facilities for both **GRM** (**General Relationship Model**) [CCITT.X.725 95] and **GDMO** (**Guidelines for the Definition of Managed Objects**) [ISO-10165.4 92] specifications. **MODE-PP TeX** is the TeX generator provided in the **MODE-PP** library.

**Mode-PP** is distributed with **Mode-FE** **FREELY** under the **COPYRIGHT** conditions as described in the **COPYRIGHT** file provided with the package. **MODE** is not a closed environment, nor it is a commercial one. Even if stable and complete, it aims at allowing people to experiment new features in Management Information Modelling.

Within this document, the features of the **MODE-PP TeX** generator are presented and the API detailed.

The remainder of this document is organised as follow. Chapter 2 provides an overview of the TeX generator's features. This overview is followed by a detailed description of the **MODE-PP TeX** generator and its use.

After some samples which illustrate some generated specifications, the document gives some information on contact and copyright concerning the **MODE** tool-set. The document is completed with a conclusion and a presentation of planned extensions (chapter 7). In an appendix, the reader will find a detailed description of the Application Programming Interface provided within the **MODE-PP TeX** as well as an example of its use.

## Chapter 2

# MODE-PP TeX overview

The MODE-PP TeX generator is the first application module built over the MODE-FE toolkit. It uses the internal representation of specifications loaded with MODE-FE and provides TeX generation methods for several specifications and/or style types.

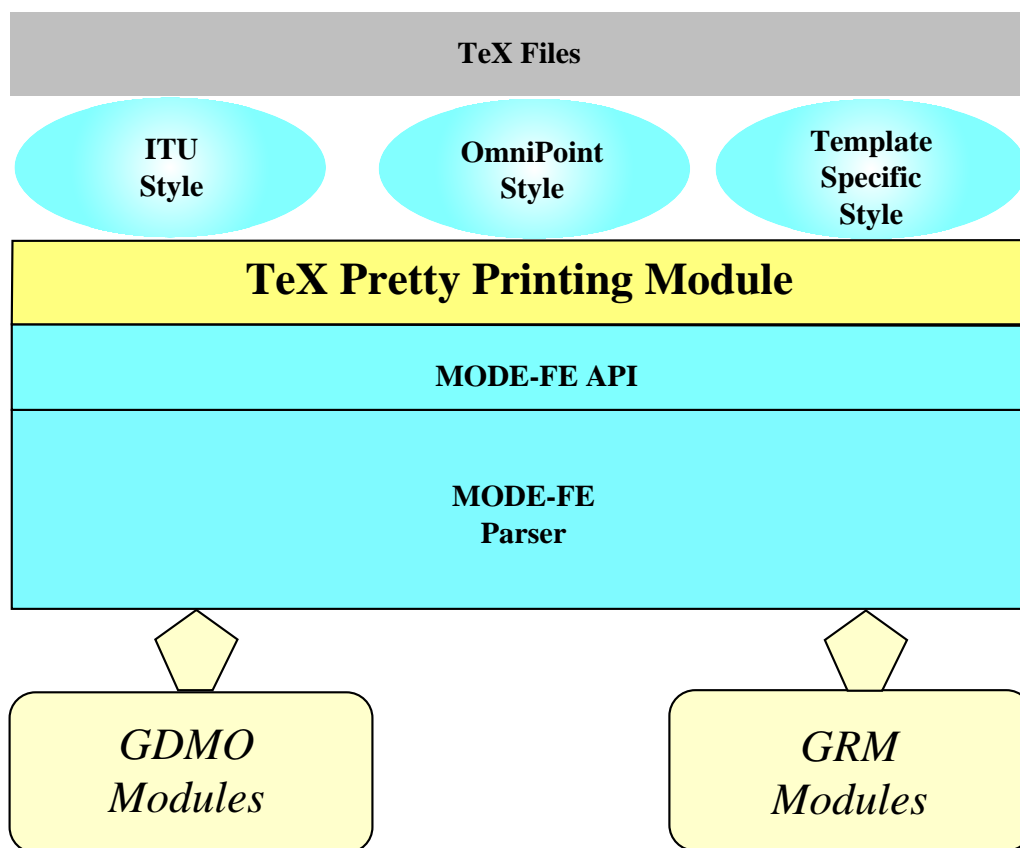


Figure 2.1: The MODE-PP Library and the TeX Generator: link to MODE-FE

The MODE-PP TeX printer offers several facilities for selecting in a very flexible way the style to which the specifications must conform. Several options are provided for this and generated TeX files are very easily integrated into any other documents. The TeX printer is provided as part of the MODE-PP library and thus provides an API allowing its integration in any other applications.

## Chapter 3

# Package Description, installation and usage

The TeX printer is provided as part of the MODE-PP library which provides several other pretty-printing facilities for GRM and GDMO specifications. The MODE-PP Library is provided as part of the MODE distribution. In its first release, the library is provided as an object code. Refer to the MODE-FE document for installation.

The full TeX printer is defined in the MODE-PP library as one C++ class with many public printing methods. These methods are detail-led in the appendix of the document. Each method is parameterised by the stream to which the output must be directed and a pointer to an object which contains the representation of the template(s) for which the method contains the TeX generator code.

Using the TeX Printer requires:

- including all Mode-FE related files,
- including the ModePP\_TexPrinter.hh file in your program,
- having an instance of the parser, the repository and the message printer,
- having an instance of the pretty printer class,
- linking your applications with both the ModeFE and the ModePP libraries.

The code below illustrates the instantiation and a call to the generation of a TeX file corresponding to a module previously parsed by the ModeFE parser.

```
// Sample Parser and module printer
#include <stdio.h>
#include <iostream.h>
#include <fstream.h>
#include "ModeFE_AllClasses.hh"
#include "ModeFE_Grammar.bison.tab.h"
#include "ModePP_TexPrinter.hh"
#include "ModePP_HTMLPrinter.hh"
#include "ModeFE_MessagePrinter.hh"

extern char* strcpy(char*);
extern "C" int strcasecmp(const char *, const char * );

parse Parser;
ModeFE_Repository repository;
ModeFE_MessagePrinter messagePrinter;
// Parser instantiation
// Repository instantiation
// message printer instantiation

int main(int argc, char **argv)
{
    FILE *fp;

    RT n° 0192
```



```

char* filename;
char* outfile;
ModePP_TexPrinter ptex;
ModePP_HTMLPrinter pout;
int erreur;
if (argc ≤ 3)
{
    cout << endl << endl << "Usage:  Main <grm-gdmo-file-name> -o <output-TeX-file> [-B]" << endl << endl;
    exit(1);
}
else
{
    filename = strcpy(argv[1]);
    if (strcasecmp(argv[2],"-o") ≠ 0)
    {
        cout << endl << "Usage:  Main <grm-gdmo-file-name> -o <output-TeX-file> [-B]" << endl << endl;
        exit(1);
    }
    else
    {
        outfile = strcpy(argv[3]);
        if ((fp = freopen(filename,"r",stdin)) == NULL)
            return ERROR ;
        else
        {
            {
                if ((erreur = Parser.yyparse()) == 0)                                     // Parser call
                {
                    ofstream OutFile ( outfile, ios::out );
                    cout << "Parsing was Successful !!!" << endl;

                    const ModeFE_Module* res = (repository.CGetModules())→CGetElement(); // module access
                    ptex.PrintOSITexModule(&OutFile, res);                               // TeX Printer call

                }
            }
            else
            {
                cout << "Parsing was unsuccessful !!!" << endl;
                cout << "...You should look at you GRM/GDMO specification!!" << endl;
            };
            fclose(fp);
        }
    }
}
}
cout << "Thanks for having used MODE-FE !\n\n\n";
};

```

## Chapter 4

# Generation samples

This chapter provides examples of generated TeX output for every type of template. This example is an excerpt from the TeX file generate for the M.3100 standard [CCITT.M.3100 92], the OSI Generic Management Information Model and the Definition of Management Information [ISO-10165.5 92, ISO-10165.2 92] and internal relationship specifications.

---

### Start Generated TeX

---

#### *Managed Object Classes*

```
network MANAGED OBJECT CLASS
  DERIVED FROM
    "CCITT Rec. X.721 (1992) — ISO/IEC 10165-2 : 1992":top;
  CHARACTERIZED BY
    networkPackage;
  CONDITIONAL PACKAGES
    userLabelPackage PRESENT IF "an instance supports it";
REGISTERED AS {ccitt recommendation m(13) gnm(3100) informationModel(0) managedObjectClass(3) 1 } ;
```

#### *Packages*

```
alarmSeverityAssignmentProfilePackage PACKAGE
  BEHAVIOUR
    alarmSeverityAssignmentProfileBehavior;
  ATTRIBUTES
    alarmSeverityAssignmentProfileId GET ,
    alarmSeverityAssignmentList GET-REPLACE ADD-REMOVE ;
;
```

#### *Attributes*

```
alarmSeverityAssignmentList ATTRIBUTE
  WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.AlarmSeverityAssignmentList ;
  BEHAVIOUR
    alarmSeverityAssignmentListBehaviour;
REGISTERED AS {ccitt recommendation m(13) gnm(3100) informationModel(0) attribute(7) 3 } ;
```

...

#### *Actions*

```
addTpsToGTP ACTION
RT n° 0192
```

**BEHAVIOUR**

addTpsToGtpBehaviour;

**MODE CONFIRMED ;****WITH INFORMATION SYNTAX** ASN1DefinedTypesModule.AddTpsToGtpInformation;**WITH REPLY SYNTAX** ASN1DefinedTypesModule.AddTpsToGtpResult;**REGISTERED AS** { ccitt recommendation m(13) gnm(3100) informationModel(0) action(9) 1 } ;*Behaviours**removeTpsFromTpPoolBehaviour* **BEHAVIOUR****DEFINED AS** "This action is used to remove termination points from termination point pools. Removing the last termination point from a pool has the effect of deleting the TP Pool object. If the TP pool is deleted, the name of the TP Pool will be sent back in the ACTION reply.";*Name-Bindings**alarmSeverityAssignment-managedElement* **NAME BINDING****SUBORDINATE OBJECT CLASS**alarmSeverityAssignmentProfile **AND SUBLCSSES;****NAMED BY SUPERIOR OBJECT CLASS**managedElement **AND SUBLCSSES;****WITH ATTRIBUTE**

alarmSeverityAssignmentProfileId;

**CREATE WITH-REFERENCE-OBJECT , WITH-AUTOMATIC-INSTANCE-NAMING ;****DELETE ONLY-IF-NO-CONTAINED-OBJECTS ;****REGISTERED AS** { ccitt recommendation m(13) gnm(3100) informationModel(0) nameBinding(6) 1 } ;*Relationship Classes**generalCompositionRelationship* **REPLATIONSHIP CLASS****BEHAVIOUR**

generalCompositionRelationshipBehaviour;

**SUPPORTS****ESTABLISH** establishGeneralComposition,**TERMINATE** terminateGeneralComposition;**ROLE** compositeRole**PERMITTED-ROLE-CARDINALITY-CONSTRAINT** GRMExample.OneToOne**REQUIRED-ROLE-CARDINALITY-CONSTRAINT** GRMExample.OneToOne**REGISTERED AS** { GRMExample . gmEx-Role x }**ROLE** componentRole**PERMITTED-ROLE-CARDINALITY-CONSTRAINT** GRMExample.OneToMax**REQUIRED-ROLE-CARDINALITY-CONSTRAINT** GRMExample.OneToOne**BIND-SUPPORT** bindComponent**BIND-SUPPORT** unbindComponent**REGISTERED AS** { GRMExample . gmEx-Role x }

;

**REGISTERED AS** { GRMExample . gmEx-RelationshipClass x } ;*Relationship Mappings**symmetricRelationshipMapping* **REPLATIONSHIP MAPPING****REPLATIONSHIP CLASS** symmetricRelationship;**BEHAVIOUR**

symmetricRelationshipMappingBehaviour;

**RELATIONSHIP OBJECT** symmetricRelationshipObject; **ROLE** peerRole **RELATED-CLASSES** "CCITT Rec. X.721 — ISO/IEC 10165-2":top**REPRESENTED BY****RELATIONSHIP-OBJECT-USING-POINTER** peer

```
; OPERATION MAPPING
  ESTABLISH establishSymmetricRelationship
MAPS-TO-OPERATION
CREATE OF RELATIONSHIP OBJECT
  TERMINATE terminateSymmetricRelationship
MAPS-TO-OPERATION
DELETE OF RELATIONSHIP OBJECT
  ESTABLISH querySymmetricRelationship
MAPS-TO-OPERATION
GET peerPointer OF RELATIONSHIP OBJECT
REGISTERED AS {GRMExample . grmEx-RelationshipMapping x } ;
```

## *Notifications*

```
attributeValueChange NOTIFICATION
  BEHAVIOUR
    attributeValueChangeBehaviour;
  WITH INFORMATION SYNTAX Notification-ASN1Module.AttributeValueChangeInfo;
  AND ATTRIBUTE IDS
    sourceIndicator sourceIndicator,
    attributeIdentifierList attributeIdentifierList,
    attributeValueChangeDefinition attributeValueChangeDefinition,
    notificationIdentifier notificationIdentifier,
    correlatedNotifications correlatedNotifications,
    additionalText additionalText,
    additionalInformation additionalInformation;
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part2(2) notification(10) 1 } ;
```

## *Attribute Group*

```
state ATTRIBUTE GROUP
  DESCRIPTION "This is defined as an empty attribute group. The elements of this group are composed of state
  attributes in the managed object. The state attributes may include those specified in CCITT Rec. X.731 — ISO/IEC
  10164-2 and others that are specific to the managed object class.";
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part2(2) attributeGroup(8) 1 } ;
```

## *Parameters*

```
miscellaneousError PARAMETER
CONTEXT SPECIFIC-ERROR ;
  WITH SYNTAX Parameter-ASN1Module.MiscellaneousError;
  BEHAVIOUR
    miscellaneousErrorBehaviour;
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part2(2) parameter(5) 1 } ;
```

---

End Generated TeX

---

## Chapter 5

# Copyright & Contact

### 5.1 Contact

The MODE toolkit is still under development. Several back-ends for MODE are either in development phase or  $\beta$ -test and some improvements to access the repository will be available in a very short time.

We are maintaining the MODE toolkit. So feel free to send us comments, bug reports. We will do our best to provide a stable and usage friendly tool.

Please send:

- bug reports,
- comments,
- any suggestions,

to one of the authors. We will respond in the shortest delay.

#### **Contact person**

Olivier Festor  
RESEDAS Research Group  
INRIA Lorraine  
Technopole de Nancy-Brabois  
- Campus scientifique -  
615, rue de Jardin Botanique - B.P. 101  
54600 Villers Les Nancy Cedex  
France  
E-mail: festor@loria.fr  
Tel: (33) 83.59.20.16  
Fax: (33) 83.27.83.19  
URL:<http://www.loria.fr/~festor>

A WWW page is maintained for MODE. The URL is:

**<http://www.loria.fr/exterieur/equipe/resedas/MODE.html>**

On this page you will find all new libraries developed for the the tool, new accessible applications, documentation, technical reports and white papers concerning planned extensions. You will also find many links to other network management pages from general information to companies which provide products for OSI and Internet Management.

### 5.2 Copyright

The ModePP library is distributed freely under the conditions specified in the COPYRIGHT file provided with the library and the MODE-FE toolkit.

## Chapter 6

# Conclusion

In this report, we have presented the TeX generation module of the MODE toolkit. This module is built over the Mode-FE parser and allows formatting of GRM and GDMO specifications. The module is in its first release still a prototype and may contain some minor errors. Future releases of the module may be provided but are not planned yet.

## Chapter 7

# Planned extensions

Several extensions are planned for the MODE-PP library. Not all are definitively retained yet. Within these extensions we already identified:

- **Specification comments treatment and "As is" formatted printing:** We are currently extending the parser in a way that the comments written in a row GRM or GDMO specification are kept in the parser and a pretty printer for raw specifications (with comments and in-line specifications) is under development.
- ...

No major extentions except those described above are planned for the TeX printing module of ModeFE.

# Bibliography

- [CCITT.M.3100 92] Comité Consultatif International Télégraphique et Téléphonique (CCITT), “*Generic Network Information Model*”, Draft International Standard, CCITT.M.3100, Janvier 1992.
- [CCITT.X.725 95] Comité Consultatif International Télégraphique et Téléphonique (CCITT), “*Information Technology - Open Systems Interconnection - Structure of Management Information - Part 7: General Relationship Model*”, Draft International Standard, CCITT.X.725, June 1995.
- [Festor 96] O. Festor, E. Nataf et L. Andrey. “MODE-FE: A GRM/GDMO Parser and its API -Release 1.0-Reference Manual”. Technical Report no. 0190, INRIA Lorraine, 1996.
- [ISO-10165.2 92] International Organization for Standardization (ISO), “*Structure of Management Information - Part 2: Definition of Management Information*”, International Standard, ISO-10165.2, January 1992.
- [ISO-10165.4 92] International Organization for Standardization (ISO), “*Structure of Management Information - Part 4: Guidelines for the Definition of Managed Objects*”, International Standard, ISO-10165.4, January 1992.
- [ISO-10165.5 92] International Organization for Standardization (ISO), “*Structure of Management Information - Part 5: Generic Management Information*”, International Standard, ISO-10165.5, January 1992.



## Appendix A

# The API

Following methods are offered by any instance of a `ModePP_TexPrinter` class.

```
ModePP_TexPrinter(int);
```

Basic constructor for the TeX pretty printing class. The parameter is used to set the maximum length of a text line in the output.

```
ModePP_TexPrinter();
```

Default constructor for the TeX pretty-printer. Creates a printer instance with a length initialized to 60 characters.

```
~ModePP_TexPrinter();
```

Destructor of the `ModePP_TexPrinter` class. Deletes all contained information.

```
ModePP_TexPrinter(const ModePP_TexPrinter&);
```

Copy constructor of the `ModePP_TexPrinter`. Does nothing in the release 1.0 of the module.

```
void PrintOSITexModule(ofstream*, const ModeFE_Module *);
```

Prints in a formatted way the module given in parameter to the stream given as a first parameter. Every information is printer to the same stream. The method generates a section for each type of template that is present in the module (managed object class, package, ...) and generates a subsection for each specification in a template type. In this subsection, the specification is formatted without a boundin-box.

The order of the printed templates is:

1. Managed Object classes,
2. Packages,
3. Attributes,
4. Attribute groups,
5. Name-Bindings,
6. Actions,
7. Notifications,
8. Parameters,
9. Relationship classes,

## 10. Relationship mappings

```
void PrintFileByFileTexModule(const ModeFE_Module *, int);
```

Generates one file per specification in the module. Each file contains the specification formatted either without a bounding-box (`int = 0`) or with a bounding box (`int = 1`). The name of the file is built from the contained specification label extended with `.tex`.

```
void PrintRelationshipMapping(ofstream*, const ModeFE_RelationshipMapping *,int );
```

Generates the Tex output which corresponds to the Relationship Mapping given in parameter. The output is send to the stream in parameter. Possible values for the type parameter (`int`) are:

- 0: no box around the specification,
- 1: the specification is encapsulated in a box.

```
void PrintRelationshipClass(ofstream*, const ModeFE_RelationshipClass *, int );
```

Generates the Tex output which corresponds to the Relationship class specification given in parameter. The output is send to the stream in parameter. Possible values for the type parameter (`int`) are:

- 0: no box around the specification,
- 1: the specification is encapsulated in a box.

```
void PrintNotification(ofstream*, const ModeFE_Notification *, int);
```

Generates the Tex output which corresponds to the notification specification given in parameter. The output is send to the stream in parameter. Possible values for the type parameter (`int`) are:

- 0: no box around the specification,
- 1: the specification is encapsulated in a box.

```
void PrintBehaviour(ofstream*, const ModeFE_Behaviour *,int );
```

Generates the Tex output which corresponds to the behaviour specification given in parameter. The output is send to the stream in parameter. Possible values for the type parameter (`int`) are:

- 0: no box around the specification,
- 1: the specification is encapsulated in a box.

```
void PrintAttributeGroup(ofstream*, const ModeFE_AttributeGroup *,int );
```

Generates the Tex output which corresponds to the attribute group given in parameter. The output is send to the stream in parameter. Possible values for the type parameter (`int`) are:

- 0: no box around the specification,
- 1: the specification is encapsulated in a box.

```
void PrintAttribute(ofstream*, const ModeFE_Attribute *,int );
```

Generates the Tex output which corresponds to the Attribute given in parameter. The output is send to the stream in parameter. Possible values for the type parameter (int) are:

- 0: no box around the specification,
- 1: the specification is encapsulated in a box.

```
void PrintNameBinding(ofstream*, const ModeFE_NameBinding *, int);
```

Generates the Tex output which corresponds to the Name-Binding given in parameter. The output is send to the stream in parameter. Possible values for the type parameter (int) are:

- 0: no box around the specification,
- 1: the specification is encapsulated in a box.

```
void PrintParameter(ofstream*, const ModeFE_Parameter *, int );
```

Generates the Tex output which corresponds to the GDMO Parameter given in parameter. The output is send to the stream in parameter. Possible values for the type parameter (int) are:

- 0: no box around the specification,
- 1: the specification is encapsulated in a box.

```
void PrintPackage(ofstream*, const ModeFE_Package *, int );
```

Generates the Tex output which corresponds to the package given in parameter. The output is send to the stream in parameter. Possible values for the type parameter (int) are:

- 0: no box around the specification,
- 1: the specification is encapsulated in a box.

```
void PrintManagedObjectClass(ofstream*, const ModeFE_ManagedObjectClass *, int );
```

Generates the Tex output which corresponds to the Managed Object Class specification given in parameter. The output is send to the stream in parameter. Possible values for the type parameter (int) are:

- 0: no box around the specification,
- 1: the specification is encapsulated in a box.

```
void PrintAction(ofstream*, const ModeFE_Action *, int );
```

Generates the Tex output which corresponds to the action given in parameter. The output is send to the stream in parameter. Possible values for the type parameter (int) are:

- 0: no box around the specification,
- 1: the specification is encapsulated in a box.

```
void PrintModule(ofstream*, const ModeFE_Module *, int);
```

Generates the Tex output corresponding to the module given in parameter to the specified stream. Possible values for the type parameter (int) are:

- 0: no box must encapsulate a specification
- 1: each specification is encapsulated in a box.

Specifications are generated in the following order:

1. Managed Object classes,
2. Packages,
3. Attributes,
4. Attribute groups,
5. Name-Bindings,
6. Actions,
7. Notifications,
8. Parameters,
9. Relationship classes,
10. Relationship mappings



---

Unit e de recherche INRIA Lorraine, Technop ole de Nancy-Brabois, Campus scientifique,  
615 rue du Jardin Botanique, BP 101, 54600 VILLERS L ES NANCY  
Unit e de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex  
Unit e de recherche INRIA Rh one-Alpes, 46 avenue F elix Viallet, 38031 GRENOBLE Cedex 1  
Unit e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex  
Unit e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

---

 diteur  
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)  
ISSN 0249-6399