

INTES

Ingenieurgesellschaft für
technische Software mbH

Breitwiesensstr.
D-70565 Stuttgart
Telefon +49 711 784 99 0

L^AT_EX at INTES

L^AT_EX 2_ε macros defined via
`\usepackage{intes}`

Internal Documentation

INTES Publication No.008, pre- α

Stuttgart, 16.3.2023

INTES

Ingenieurgesellschaft für
technische Software mbH

Breitwiesenstr. 28
D-70565 Stuttgart
Telefon +49 711 784 99 - 0

Front Page Example

Using Macro \titlep

Front Page Example

Macro INTES**TITLE**

N.N.

PERMAS Version 5.01

INTES Publication No.007, Rev.pre- α

Stuttgart, 1995

INTES

Ingenieurgesellschaft für
technische Software mbH
Breitwiesenstr. 28
D-70565 Stuttgart
Tel. +49 711 784 99 - 0

Copyright 2023 by INTES GmbH, Stuttgart, Germany.

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, including photographing, recording and storage in a data processing system without written permission by INTES GmbH.

PERMAS and VisPER are registered trademarks by INTES GmbH.

The use of registered names or trademarks, etc. in the publication does not imply, even in the absence of further specific statements, that such names are exempt from the relevant legal protection and free for general use.

Das Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, der Entnahme von Abbildungen, der Wiedergabe auf photomechanischem oder ähnlichem Weg und der Speicherung in Datenverarbeitungsanlagen bleiben, auch bei auszugsweiser Verwertung, vorbehalten.

PERMAS und VisPER sind eingetragene Warenzeichen der INTES GmbH.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Marktschutz-Gesetzgebung als frei zu betrachten sind und daher von jedermann benutzt werden dürften.

- 1) Abaqus is a registered trademark of Dassault Systèmes Simulia Corp., Providence, RI, USA.
- 2) Adams is a registered trademark of MSC Software Corporation, Santa Ana, USA.
- 3) ADSTEFAN is a registered trademark of Hitachi Industry Control & Solutions, Ltd., Ibaraki, Japan.
- 4) AVL Excite is a registered trademark of AVL List GmbH, Graz, Austria.
- 5) COMREL is a registered trademark of RCP GmbH, München, Germany.
- 6) DADS is a registered trademark of LMS CADSI, Coralville, USA.
- 7) FEMFAT is a registered trademark of Engineering Center Steyr GmbH & Co KG, St. Valentin, Austria.
- 8) HyperMesh, HyperView, H3D and MotionSolve are registered trademarks of the Altair Engineering Inc., Big Beaver, USA.
- 9) I-DEAS is a registered trademark of SIEMENS PLM Software Inc., Plano, USA.
- 10) Java is a registered trademark of Sun Microsystems, Inc. in the United States and other countries.
- 11) MATLAB is a registered trademark of The Mathworks Inc., Natick, MA, USA.
- 12) MEDINA and CAE-Datenschiene are registered trademarks of the T-Systems ITS GmbH, Stuttgart, Germany.
- 13) MeshWorks is a registered trademark of Detroit Engineered Products, Troy, Michigan, USA.
- 14) MotionSolve is a registered trademark of Altair Engineering Inc., Big Beaver, USA.
- 15) MpCCI is a registered trademark of FhG SCAI, St. Augustin, Germany.
- 16) docx, xlsx, pptx are registered trademarks of the Microsoft Corporation, Redmond, WA, USA.
- 17) MSC Nastran is a registered trademark of MSC Software Corporation, Santa Ana, USA.
- 18) Nastran is a registered trademark of the National Aeronautics and Space Administration (NASA).
- 19) NX is a registered trademark of Siemens PLM Software.
- 20) MSC Patran is a registered trademark of MSC Software Corporation, Santa Ana, USA.
- 21) Python is a registered trademark of the Python Software Foundation.
- 22) Sun Grid Engine is a registered trademark of Sun Microsystems, Inc., Palo Alto, CA, USA.
- 23) Simdrive 3D is a registered trademark of CONTECS Engineering Services GmbH, Berlin, Germany.
- 24) SIMPACK is a registered trademark of Dassault Systèmes Simulia Corp., Providence, RI, USA.
- 25) STAR CD is a registered trademark of Computational Dynamics Ltd., London, England.
- 26) TecWare is a registered trademark of LMS International, Leuven, Belgium.
- 27) TOSCA is a registered trademark of FE-DESIGN GmbH, Karlsruhe, Germany.
- 28) VAO is a registered trademark of CDH AG, Ingolstadt, Germany.
- 29) Virtual.Lab is a registered trademark of LMS International, Leuven, Belgium. **No additional trademarks !**

The list does not claim to be exhaustive.

A list of used 3rd party libraries can be found in the respective *Installation and Operations Manual* UM909/UM910.

Preface

Dies ist der Versuch alle L^AT_EX-Makros zu beschreiben, die auf der APOLLO im File

`//node_cd21/arbortext/inputs/[g]intes.sty`

enthalten sind. All diese Makros sind zusätzlich zu den normalen L^AT_EX-Befehlen verfügbar, sobald am Anfang eines T_EX-Dokumentes als `documentstyle` z.B. `\documentstyle{intes}` oder `\documentstyle{intes,11pt,makeindex}` {a o.ä. eingefügt wird.

Alternativ kann der T_EX-File statt mit `latex[egf] myfile[.tex] [onlyfile]` auch mit `intex[egf] myfile[.tex] [onlyfile]` bearbeitet werden (geht etwas schneller) — dann darf natürlich `intes` nicht innerhalb des Documentstyles verwendet werden.

Die vorliegende Beschreibung steht in dem APOLLO-Directory

`//node_cd21/internal_doc/intes_style` (Hauptfile `intes.tex`)

und kann dort von jedem ergänzt werden, der sich dazu berufen fühlt. Insbesondere wird darum gebeten, daß jeder, der neue Makros in den *intes.sty* einfügt, diese auch hier dokumentiert und eventuell mit einem kleinen Beispiel versieht.

This document is subject to modification without prior notice !

P.S.: Bitte alle gefundenen Fehler oder Unstimmigkeiten – auch in der vorliegenden Beschreibung – bei den Autoren melden (unknown bugs will stay and stay and ...) !

New Features and Previous Revisions

- Rev.A : Anlaß zur ersten Dokumentation war die Einführung neuer Environments zur Beschreibung von DMS-Datenstrukturen (*Describing Data Structures* in Kapitel 6).
- Rev.B : Erweiterte Beschreibung älterer Makros und Beseitigung kleinerer Fehler.
- Rev.C : Neue Environments zur Beschreibung von Algorithmen und Programm-Modulen (Kapitel 7, *Describing Algorithms*) sowie kleinere Utilities im Abschnitt 4.5 (\B...E).
- Rev.D : Neue Utilities im Kapitel 4 (*General Utilities*): \titel...TITEL (4.2.4), \head (4.2.1), \verbatimfile bzw. \verbatimlisting (4.4). Als weiteres Page Size Makro wurde \fullpage aufgenommen (3.2.3). Im Kapitel 7.5 (*Describing Algorithms*) wurde \call modifiziert und dafür ein zusätzliches \Call eingeführt.
- Rev.E : Komplette Überarbeitung des *intes.sty* ! Aufgrund neuester Erkenntnisse in der hausinternen T_EX-Forschung konnte der Style-File von fast 2000 Zeilen auf unter 800 Zeilen gekürzt werden — bei gleichzeitig erweitertem Funktionsumfang (vor allem die DMS-Environments aus Kapitel 6 wurden umgeschrieben und z.B. um \DMScolsep erweitert).
Im Kapitel 2.1 werden kleinere Makros vorgestellt, die eine Aufteilung der T_EX-Files in Hauptdokument und Teildokumente vereinfachen (*DOC-Utilities*, siehe auch Abschnitt 1.2). Im Kapitel 3 (*Page Layout*) wurde das Front-Page Makro \INTESTITLE um ein 5.Argument erweitert und \trademarks als eigenständiges Kommando eingeführt. Auch die Möglichkeiten zur Index-Erzeugung wurden durch kleinere Makros verbessert (z.B. fett nummerierte Haupteinträge), siehe Kapitel 10 (*Index Making*).
Auf APOLLO besteht nun die Möglichkeit durch `intex myfile[.tex]` anstelle von `latex myfile[.tex]` die Intes-Makros direkt zu laden (und nicht über \documentstyle erst zu interpretieren). Last but not least wurde die vorliegende Beschreibung etwas umgestellt und um das jetzige Kapitel 1 sowie mehrere Abschnitte und einige Anhänge erweitert.
- Rev.F : Das Kapitel 7 (*Describing Algorithms*) wurde komplett überarbeitet ! Ähnlich wie die DMS-Environments in Rev.E wurden diese Makros vereinheitlicht, verkürzt und gleichzeitig stark verbessert. Um bei stark verschachtelten Environments nicht an die T_EX-Grenze *TeX capacity exceeded [main memory size 65536]* zu stoßen wurde mit BEcheckOFF die Möglichkeit geschaffen den L^AT_EX-Environment-Stack vorübergehend abzuschalten (4.5).
Im Kapitel 5 wurden die schon seit längerem vorhandenen List-Environments ARGDES, ARGLIST und arg\$ beschrieben und in 6 *Describing Data Structures* kamen die Makros \Newtup und \Newlist hinzu. Ab sofort kann makeidx im \documentstyle entfallen, da dessen Definition bereits im *intes.sty* integriert ist, siehe Anhang E (MakeIndex). Bei \DOCcontents (2.2) wurde die Einrücktiefe an mehrzifferige Abschnittsnummerierungen angepaßt und durch \docsec (3.4) werden jetzt auch die Figures und Tables kapitelweise gezählt. Ferner wurden in Abschnitt 4.1 einige neue Symbole definiert.
- Rev.G : Makro-Definitionen abgeschlossen, Dokumentation aber noch unvollständig !
Wichtigste Neuerung: Grafik ohne Schere und Kleber → Konvertierungsprogramm pcl für HPGL-Files im Anhang und darauf aufbauende PCL-Makros.
Bei *Describing Data Structures* wurde die neue List Naming Rule \lisnam sowie das Zusatzmakro \Addlin eingeführt (Abschnitt 6.9 bzw. 6.8). Mit Hinblick auf Kapitel 5 (*Describing Functions and Subroutine*) kamen im 4.Kapitel die *General Utilities* \uhead und \I bzw. \U hinzu (4.2.2 und 4.6.2), siehe auch 5.6.
Das Gleiten von L^AT_EX-Floats wurde verbessert: \LaTeXfloats und \DOCfloats.
Zur Ergänzung der *DOCutilities*: Pagestyle \DOCint und stark erweiterte Standardumgebung \begin{DOC} ... \end{DOC}. Auf \begin{DOC} basierende Möglichkeit echte Kreuzreferenzen zwischen verschiedenen Dokumenten zu erzeugen (XRF-Makros 2.3) bis hin zur Erstellung eines globalen Indexverzeichnisses über mehrere Dokumente.

Contents

Preface	i
1 Introduction	1-1
2 Standardized Documentation	2-1
3 Page Layout	3-1
4 General Utilities	4-1
5 Describing Functions and Subroutines	5-1
6 Describing Data Structures	6-1
7 Describing Algorithms	7-1
8 Describing Syntax Structures	8-1
9 Bibliographic References	9-1
10 Index Making	10-1
A Calling TeX: <code>intex[g] myfile[.tex] [onlyfile]</code>	A-1
B Output to Screen: <code>preview myfile[.dvi]</code>	B-1
C Creating Output for HP-Laser: <code>dvihp myfile[.dvi]</code>	C-1
D Printing HP-Laser Files: <code>laser myfile[.hp]</code>	D-1
E <code>makeindex myfile[.idx]</code>	E-1
Index	I

Table of Contents

Table of Contents

Preface	i
New Features and Previous Revisions	ii
1 Introduction	1-1
1.1 Using INTES Macros	1-1
1.2 Partitioning LaTeX-Files	1-2
1.2.1 Main Document Example 1	1-2
1.2.2 Main Document Example 2	1-3
1.3 Additional Hints to German Styles	1-4
2 Standardized Documentation	2-1
2.1 Global Document Environment: <code>\begin{DOC}[3]...\end{DOC}</code>	2-1
2.2 DOCutilities	2-2
2.2.1 Improved Float Parameter: <code>\DOCfloat</code> and <code>\LaTeXfloat</code>	2-2
2.2.2 Preceding Text: <code>\DOCabstract</code> and <code>\DOCpreface</code>	2-2
2.2.3 Listing Contents: <code>\DOCcontents</code> , <code>\DOCfigures</code> and <code>\DOCtables</code>	2-2
2.2.4 Starting of Text: <code>\DOCtext</code> and <code>\DOCnumbering[1]</code>	2-2
2.2.5 Start of Appendices: <code>\DOCappendix</code>	2-3
2.2.6 Concluding Index: <code>\DOCindex</code>	2-3
2.3 Cross References To and From other Documents	2-4
2.3.1 Search Path for XRF-Files: <code>\XRFinputpath[1]</code>	2-4
2.3.2 Including XRF-Files from other Documents: <code>\XRFinput[1]</code>	2-4
2.3.3 XRF-Identificator, -Title and -Revision: <code>\XRFtitle[1]</code> and <code>\XRFrev[1]</code>	2-4
2.3.4 Export of Labels: <code>\XRFlabel[1]</code>	2-4
2.3.5 Import of Labels: <code>\XRFref[2]</code> and <code>\XRFpageref[2]</code>	2-4
2.3.6 Cross Reference Index: <code>\XRFindex[1]</code>	2-5
3 Page Layout	3-1
3.1 Front Pages	3-1
3.1.1 Front Page Header: <code>\inteskopf</code>	3-1
3.1.2 Internal Documents: <code>\internaldoc[5]</code>	3-1
3.1.3 General INTES Document: <code>\titlep[2]</code>	3-1
3.1.4 External Documents: <code>\INTESTITLE[6]</code>	3-2
3.1.5 Dynamic Titeling: <code>\titel...\TITEL</code>	3-2

3.1.6	Trademarks and Copyrights: <code>\trademarks[1]</code>	3-2
3.2	Page Size	3-3
3.2.1	<code>\DINA4</code>	3-3
3.2.2	<code>\docpage</code>	3-3
3.2.3	<code>\fullpage</code>	3-3
3.3	Page Styles	3-3
3.3.1	Internal Documentation: <code>\pagestyle{docint}</code>	3-3
3.3.2	Internal Standard Documentation: <code>\pagestyle{DOCint}</code>	3-5
3.3.3	Auftragsprotokoll: <code>\pagestyle{AUFTRAG}</code>	3-7
3.4	Document Sectioning	3-7
3.4.1	Document Section: <code>\docsec[1]</code>	3-7
4	General Utilities	4-1
4.1	Additional Symbols	4-1
4.2	Emphasized Text for Headings, Titles or Remarks	4-1
4.2.1	Paragraph Header: <code>\head[1]</code>	4-1
4.2.2	Underlined Paragraph Header: <code>\uhead[1]</code>	4-2
4.2.3	Paragraph Header (Big and Bold): <code>\bigbf[1]</code>	4-2
4.2.4	Dynamic Titeling: <code>\titel...\TITEL</code>	4-3
4.3	Note and Boxes	4-4
4.3.1	Note with Top and Bottom Line: <code>\note[1]</code>	4-4
4.3.2	Text Box: <code>\Kasten[1]</code>	4-4
4.3.3	Empty Box: <code>\Rahmen[1]</code>	4-4
4.4	Extensions to the <code>\verbatim</code> Environment	4-5
4.4.1	Include File as ASCII-Text: <code>\verbatimfile[1]</code>	4-5
4.4.2	Include File as Listing: <code>\verbatimlisting[1]</code>	4-5
4.5	Short Commands <code>\beg</code> , <code>\B</code> and <code>\E</code> for Environments	4-7
4.5.1	Dirty Tricks with <code>\begin{BEcheckOFF}...\end{BEcheckOFF}</code>	4-7
4.6	More Utilities	4-8
4.6.1	Actual Date: <code>\heute</code>	4-8
4.6.2	Shortened Italic and Underline: <code>\I[1]</code> and <code>\U[1]</code>	4-8
4.6.3	Absolut Value: <code>\abs[1]</code>	4-8
4.6.4	Dynamic Horizontal Spacing: <code>\leer...\LEER</code>	4-8
4.6.5	Bold List item: <code>\bfitem[1]</code>	4-8
4.7	Environments for Listing Definitions	4-9

Table of Contents

4.7.1	Left Adjusted Items: <code>\B{deflist}...\E{deflist}</code>	4-9
4.7.2	Right Adjusted Items: <code>\B{listdef}...\E{listdef}</code>	4-9
5	Describing Functions and Subroutines	5-1
5.1	Definition of Function/Subroutine: <code>\B{Call}[3]...\E{Call}</code>	5-1
5.2	List of Arguments: <code>\B{arg1...}...\End{arg1...}</code>	5-2
5.3	Description of Arguments: <code>\B{argd...}...\E{argd...}</code>	5-3
5.4	Argument Item with Array Counter: <code>\ARGi[2]</code>	5-5
5.5	List of Errors: <code>\B{errlist}...\E{errlist}</code>	5-6
5.6	Summary Example	5-7
6	Describing Data Structures	6-1
6.1	Fortran Vector: <code>\B{DMSvector}...\E{DMSvector}</code>	6-1
6.2	Fortran Array: <code>\B{DMSarray}[2]...\E{DMSarray}</code>	6-2
6.3	DMS Table: <code>\B{DMStable}[2]...\E{DMStable}</code>	6-5
6.4	DMS Control Table: <code>\B{DMSCtable}...\E{DMSCtable}</code>	6-9
6.5	DMS Sequence: <code>\B{DMSsequence}[2]...\E{DMSsequence}</code>	6-11
6.6	DMS Control Sequence: <code>\B{DMSCsequence}...\E{DMSCsequence}</code>	6-13
6.7	DMS List: <code>\B{DMSlist}[1]...\E{DMSlist}</code>	6-15
6.8	DMS Control List: <code>\B{DMSClist}...\End{DMSClist}</code>	6-17
6.9	Naming Rule for List Names: <code>\namlis[3]</code>	6-19
6.10	Title for Textual Description of Data Objects: <code>\DMStitle[3]</code>	6-19
6.11	DMS Book: <code>\B{DMSbook}[4]...\E{DMSbook}</code>	6-20
6.12	Local Redefinition of DMStype for Help-Files: <code>\DMStype[1]</code>	6-20
7	Describing Algorithms	7-1
7.1	Module Environment: <code>\begin{Module}[2]...\end{Module}</code>	7-1
7.2	Separating Blocks: <code>\begin{Block}...\end{Block}</code> and <code>\block[1]</code>	7-3
7.3	Definition of Loops: <code>\begin{Loop}[1]...\end{Loop}</code> and <code>\loopblock[2]</code>	7-4
7.4	Case Structures: <code>\begin{If}...\end{If}</code> and <code>\ifblock[2]</code>	<code>\begin{Case}...\end{Case}</code> and <code>\caseblock[2]</code>
7.5	Referring to Submodules: <code>\call[1]</code> and <code>\Call[1]</code>	7-10
7.6	Summary Example	7-11
8	Describing Syntax Structures	8-1
8.1	Syntax Environment: <code>\begin{Syntax}[3]...\end{Syntax}</code>	8-1
9	Bibliographic References	9-1

Table of Contents

10 Index Making	10-1
10.1 Definition of Index Entries	10-1
10.1.1 The Basics: <code>\index[1]</code>	10-1
10.1.2 Primary Index: <code>\indexbf[1]</code>	10-2
10.1.3 Print and Create an Index: <code>\Index[1]</code>	10-2
10.1.4 List Item and Create an Index: <code>\Item[1]</code>	10-2
10.2 Index Layout	10-2
A Calling TeX: <code>intex[g] myfile[.tex] [onlyfile]</code>	A-1
B Output to Screen: <code>preview myfile[.dvi]</code>	B-1
C Creating Output for HP-Laser: <code>dvihp myfile[.dvi]</code>	C-1
C.1 DVILASER Options	C-1
C.2 Interactive Options	C-2
D Printing HP-Laser Files: <code>laser myfile[.hp]</code>	D-1
E <code>makeindex myfile[.idx]</code>	E-1
E.1 How to Use <i>MakeIndex</i>	E-1
Index	I

1 Introduction

1.1 Using INTES Macros

Um INTES Makros benutzen zu können, muß entweder im `\documentstyle[...]` zusätzlich `intes` mit angegeben werden, oder mit `intex myfile[.tex]` übersetzt werden.

Beides geht nicht \longrightarrow doppelte Definitionen

Falls alte \LaTeX -Files mit der Revision *pre- α* des *intes.sty* nicht mehr zu übersetzen sind und kein \LaTeX Profi im Hause ist, kann man, um weiterzuarbeiten, einen älteren Style-File laden. z.B.: `\documentstyle[intesE,...]`. Nun kann man den \LaTeX -File in gewohnter Weise mit `latex myfile[.tex]` übersetzen.

1.2 Partitioning LaTeX-Files

1.2.1 Main Document Example 1

```
\documentstyle[11pt]{article} % 11pt-Schriften
\DINA4                        % Seitengroesse
\makeindex                    % Indexfile erzeugen
\begin{document}
%
% =====
\internaldoc {Documenttitle}{Subtittle}
              {Author}{No.007,Rev.A}{\heute}
%
% =====
\pagestyle{docint} % Pagestyle waehlen
%
\DOCpreface % Jetzt kommt das Vorwort
\input vorwort % TeX-File (ohne \docsec) mit Vorwort
%
\DOCcontents % Inhaltsverzeichnis
\DOCfigures % List of figures
\DOCTables % List of tables
\DOCtext % Normale Seitenzaehlung
%
\input teil_1 % TeX-File mit 1.Kapitel (Titel in \docsec)
% -----
%
\input teil_2 % TeX-File mit 2.Kapitel (Titel in \docsec)
% -----
%
\input teil_3 % TeX-File mit 3.Kapitel (Titel in \docsec)
% -----
%
\DOCappendix % Jetzt kommt der Anhang:
\input teil_A % TeX-File mit 1.Anhang (Titel in \docsec)
\input teil_B % TeX-File mit 2.Anhang (Titel in \docsec)
%
\DOCindex % Index
%-----
\end{document}
```

1.2 Partitioning LaTeX-Files

1.2.2 Main Document Example 2

```
\input intes_ini          % TeX-Initialisierung
\def\Rev{F+}              % Revision number (multiple used)
%-----
\begin{DOC}{INTEX}{LaTeX at INTES}{Rev.\Rev} %%% \LaTeX\ at INTES}
%-----
\include{intes_tit}       % Title
\include{intes_pre}       % Preface
\include{intes_toc}       % Table of Contents
%
\DOCtext                  % Normale Seitenzaehlung
\include{intes_int}       % Introduction
\include{intes_doc}       % Standardized Documentation
\include{intes_page}      % Page Layout
\include{intes_util}      % General Utilities
\include{intes_sub}       % Describing Functions and Subroutines
\include{intes_dat}       % Describing Data Structures
\include{intes_alg}       % Describing Algorithms
\include{intes_graph}     % Including Graphics
\include{intes_bib}       % Bibliographic References
\include{intes_index}     % Index Making
%
\DOCappendix              % Appendix:
\include{intes_tex}       % LaTeX, INTEX Commands
\include{intes_preview}% PREVIEW Commands
\include{intes_dvihp}     % DVIHP Commands
\include{intes_laser}     % LASER Commands
\include{intes_makeidx}% Make Index
\include{intes_pcl}       % PCL-Merging of HPGL Files
%
\include{intes_ind}       % Index of document
%-----
\end{DOC}
```

1.3 Additional Hints to German Styles

`\documentstyle[german,intes ...]` oder `\documentstyle[intes,german ...]`

ermöglicht die deutschen Umlaute	ä	ö	ü	Ä	Ö	Ü	ß
statt mit den T _E X-Hieroglyphen	<code>\"a</code>	<code>\"o</code>	<code>\"u</code>	<code>\"A</code>	<code>\"O</code>	<code>\"U</code>	<code>\ss</code>
etwas besser lesbar als	<code>"a</code>	<code>"o</code>	<code>"u</code>	<code>"A</code>	<code>"O</code>	<code>"U</code>	<code>"s</code>
zu beschreiben.							

Alternativ dazu gibt es noch die Möglichkeiten *german format files* zu benutzen:

- 1 : Durch den Aufruf `latex myfile[.tex]` anstelle von `latex myfile[.tex]`.
- 2 : Durch den Aufruf `intex myfile[.tex]` anstelle von `intex myfile[.tex]`.
- 3 : Durch das Laden von *german styles*
 - a : *garticle* statt *article*
 - b : *greport* statt *report*
 - c : *gbook* statt *book*
 - d : usw.

Bei der Benutzung von German Styles in Verbindung mit `\DOC. . .` wird für foldende Makros die Kapitelüberschrift in deutsch gesetzt:

- `\DOCpreface` : Anstelle *Preface* wird *Vorwort* als Kapitelüberschrift gewählt. Dieses wird ebenfalls in das Inhaltsverzeichnis und in die Seitenüberschrift des Pagestyles `\pagestyle{DOCint}` eingetragen.
- `\DOCabstract` : Anstelle *Abstract* wird *Übersicht* als Kapitelüberschrift gewählt. Dieses wird ebenfalls in das Inhaltsverzeichnis und in die Seitenüberschrift des Pagestyles `\pagestyle{DOCint}` eingetragen.
- `\DOCcontents` : Anstelle *Contents* wird *Inhaltsverzeichnis* als Kapitelüberschrift gewählt. Dieses wird ebenfalls als Seitenüberschrift des Pagestyles `\pagestyle{DOCint}` eingetragen.
- `\DOCfigures` : Anstelle *Figures* wird *Abbildungsverzeichnis* als Kapitelüberschrift gewählt. Dieses wird ebenfalls als Seitenüberschrift des Pagestyles `\pagestyle{DOCint}` eingetragen.
- `\DOCtables` : Anstelle *Tables* wird *Tabellenverzeichnis* als Kapitelüberschrift gewählt. Dieses wird ebenfalls als Seitenüberschrift des Pagestyles `\pagestyle{DOCint}` eingetragen.

2.1 Global Document Environment: `\begin{DOC}[3]...\end{DOC}`

2 Standardized Documentation

Empfehlung: Pagestyle DOCint, DOC, . . .

2.1 Global Document Environment: `\begin{DOC}[3]...\end{DOC}`

Usage : `\begin{DOC}{xrfid}{xrftitle}{xrfrev}`

Items : `xrfid` : XRF-Identificator
 `xrftitle` : XRF-Title
 `xrfrev` : XRF-Revision

`\makeindex` und `\DOCfloats` integriert. Hinweis auf pagestyle DOCint und XRF . . .

Nur mit DOC: `\XRF...`, `[onlyfile]` in `latex/intex` call !

Besser mit DOC: Pagestyle DOCint, kein `\makeindex` mehr nötig !

Mit/ohne DOC gleich: Alle sonstigen Makros, inklusive DOCutilities !

2.2 DOCutilities

The following macros do not depend on the `\begin{DOC}... \end{DOC}` environment !

2.2.1 Improved Float Parameter: `\DOCfloat` and `\LaTeXfloat`

`\DOCfloat`
`\LaTeXfloat`

2.2.2 Preceding Text: `\DOCabstract` and `\DOCpreface`

Die Vorworte werden in das Inhaltsverzeichnis eingetragen, erhalten jedoch keine Kapitelnummern. Die Seitenzählung ist fortlaufend in `\roman`.

`\DOCabstract` Erstellt ein Vorwort mit der Überschrift und dem Seitenkopf *Abstract* (bei Verwendung von German-Styles *Übersicht*).
`\DOCpreface` Erstellt ein Vorwort mit der Überschrift und dem Seitenkopf *Preface* (bei Verwendung von German-Styles *Vorwort*).

2.2.3 Listing Contents: `\DOCcontents`, `\DOCfigures` and `\DOCtables`

Die Verzeichnisse erhalten keine Kapitelnummern und werden nicht im Inhaltsverzeichnis aufgeführt. Die Seitenzählung ist fortlaufend in `\roman`.

`\DOCcontents` Erstellt ein Inhaltsverzeichnis mit der Überschrift und dem Seitenkopf *Contents* (bei Verwendung von German-Styles *Inhaltsverzeichnis*). Die Einrückung ist an mehrziffrige Abschnitte angepaßt.
`\DOCfigures` Erstellt ein Bildverzeichnis mit der Überschrift und dem Seitenkopf *Figures* (bei Verwendung von German-Styles *Abbildungsverzeichnis*).
`\DOCtables` Erstellt ein Tabellenverzeichnis mit der Überschrift und dem Seitenkopf *Tables* (bei Verwendung von German-Styles *Tabellenverzeichnis*).

2.2.4 Starting of Text: `\DOCtext` and `\DOCnumbering[1]`

Die Seitennummerierung ist jeweils Kapitelzähler – Seitenzähler.

`\DOCtext` Setzt die Kapitelzählung und die Seitenzählung auf `\arabic`.
`\docsec` Inkrementiert den Kapitelzähler (`section`) und beginnt ein neues Kapitel auf neuer Seite. Der Seitenzähler und die Zähler der Bilder (`figures`) und Tabellen (`tables`) Umgebungen werden zurückgesetzt (x-0), siehe Abschnitt 3.4 auf 3-7.
`\DOCnumbering{3}` Setzt Kapitelzähler auf eine Zahl, z.B.: für das Erstellen von Dokumententeilen.

2.2 DOCutilities

2.2.5 Start of Appendices: `\DOCappendix`

`\DOCappendix` Setzt die Kapitelzählung auf `\Alph` und setzt die Seitenzählung auf `\arabic`.

2.2.6 Concluding Index: `\DOCindex`

`\DOCindex` Erstellt einen Index mit der Überschrift und dem Seitenkopf Index. Der Index erhält keine Kapitelnummer und wird in das Inhaltsverzeichnis eingetragen. Die Seitenzählung ist fortlaufend in `\Roman`.

2.3 Cross References To and From other Documents

The following macros will work with `\begin{DOC}...\end{DOC}` environment only !

2.3.1 Search Path for XRF-Files: `\XRFinputpath[1]`

Search-path for XRF-Input-Files (e.g. in preamble or directly before `\begin{DOC}` respectively before `\XRFinput`):

Usage	: <code>\XRFinpath{pathname}</code>	% Optional
Items	: <code>pathname</code> : HP search path for XRF-Files, e.g. <code>/support/doc/xrf/</code> or <code>../xrffiles/</code> (terminating slash is important), default is an empty <code>pathname</code> , i.e. the actual directory.	

2.3.2 Including XRF-Files from other Documents: `\XRFinput[1]`

Usage	: <code>\XRFinput{xrfname}</code>
Items	: <code>xrfname</code> : Name of XRF-File (without extension).

2.3.3 XRF-Identificator, -Title and -Revision: `\XRFTitle[1]` and `\XRFFrev[1]`

Usage	: <code>\XRFTitle{xrfid}</code>	and	<code>\XRFFrev{xrfid}</code>
Items	: <code>xrfid</code> : XRF-Identificator of imported document.		

2.3.4 Export of Labels: `\XRFLabel[1]`

Usage	: <code>\XRFLabel{xrflabel}</code>
Items	: <code>xrflabel</code> : XRF-Label to be exported.

2.3.5 Import of Labels: `\XRFref[2]` and `\XRFpageref[2]`

Usage	: <code>\XRFref{xrfid}{xrflabel}</code>	and	<code>\XRFpageref{xrfid}{xrflabel}</code>
Items	: <code>xrfid</code> : XRF-Identificator of imported document.		
	: <code>xrflabel</code> : XRF-Label to be imported.		

2.3 Cross References To and From other Documents

2.3.6 Cross Reference Index: `\XRFindex[1]`

Usage : `\XRFindex{xrfindex}`

Items : `xrfindex` : XRF-Index to be exported.

3.1 Front Pages

3 Page Layout

3.1 Front Pages

3.1.1 Front Page Header: `\inteskopf`

`\inteskopf` erzeugt einen INTES-Header für Titelseiten, so wie er auch für das vorliegende Dokument verwendet wurde (vgl. auch `\internaldoc`).

3.1.2 Internal Documents: `\internaldoc`[5]

`\internaldoc` erzeugt eine Titelseite für eine interne Dokumentation (mit `\inteskopf`):

Usage : `\internaldoc{title}{subtitle}{author}{NrRev}{date}`

Items : title : Document title
subtitle : Subtitle (can be empty {})
author : Author(s)
NrRev : Publication number and revision in the form
No.xxx, Rev.A or equivalent specification.
date : Date of publication, e.g. year

Example : The first front page was defined via:

```
\internaldoc { \LaTeX\ at INTES (Out-Of-Date!)}
              {\LaTeX\ macros defined via \
               {\tt intex[g] myfile[.tex] [onlyfile]}\
               or via \LaTeX documentstyle\
               {\tt$\backslash$documentstyle[intes\hdots]}}
              {Markus Ast \ Hans Storm}
              {No.007, Rev.\Rev} {\heute}
```

3.1.3 General INTES Document: `\titlep`[2]

`\titlep` erzeugt eine Titelseite mit INTES Kopf (etwas fetter als bei `\inteskopf`):

Usage : `\titlep{title}{author}`

Items : title : Document title
author : Author(s)

Example : The second front page was defined via:

```
\titlep {Front Page Example}
        {Using Macro $\backslash$titlep}
```

3.1.4 External Documents: `\INTESTITLE[6]`

`\INTESTITLE` erzeugt ein zweiseitiges Titelblatt mit Copyright und allem was so dazugehört (für die *INTES-Loch-Umschläge*):

Usage : `\INTESTITLE{title}{author}{PrgV}{NrRev}{date}{trademarks}`

Items :

- title : Document title
- author : Author(s)
- PrgV : Program version, e.g. [PERMAS Version 3.1]
- NrRev : Publication number and revision in the form
No. xxx, Rev. A or equivalent specification.
- date : Date of publication, e.g. year
- trademarks : Additional trademarks, can be empty {}
(benutzt intern das `\trademarks` Makro).

Example : The front pages 3 & 4 were defined via:

```
\INTESTITLE {Front Page Example \}[10mm]
Macro INTESTITLE}
{N.N.}
{PERMAS \ Version 3.1}
{No.007, Rev.pre- $\alpha$ } {1990}
{{\bf No additional trademarks !}}
```

3.1.5 Dynamic Titeling: `\titel...\TITEL`

Various titles from paragraph up to front page titeling, see section 4.2.4 on page 4-3.

3.1.6 Trademarks and Copyrights: `\trademarks[1]`

Mit `\trademarks{Additional trademarks}` kann nach jeder beliebigen Titelseite noch ein INTES-Copyright wie bei `\INTESTITLE` eingefügt werden (vgl. 4.Seite dieses Dokuments).

3.2 Page Size

3.2.1 `\DINA4`

Definiert die Größe der deutschen DIN A4 Seite

3.2.2 `\docpage`

Definiert eine Seitengröße die etwas kleiner als `\DINA4` ist.

3.2.3 `\fullpage`

Nutzt alles aus was geht !

Kein Seitenkopf, -fuß, keine Seitenzählung, kein Randbereich — nicht für Dokumente !

3.3 Page Styles

3.3.1 Internal Documentation: `\pagestyle{docint}`

Das Kommando `\pagestyle{docint}` definiert einen L^AT_EX-Pagestyle wie er auch für diese Seite verwendet wurde. Bei zweiseitigem Druck gilt:

Left page header:

Kapitel-Seite

Kapitelnummer Überschrift

Right page header:

Unterkapitelnummer Überschrift

Kapitel-Seite

Wird einseitig gedruckt (wie bei diesem Dokument), so bewirkt `\pagestyle{docint}`:

Page header:

Kapitelnummer Überschrift

Kapitel-Seite

Der Fußbereich bleibt – im Gegensatz zum folgenden Pagestyle `DOCint` – vollständig leer !

3.3 Page Styles

3.3.2 Internal Standard Documentation: `\pagestyle{DOCint}`

Das Kommando `\pagestyle{DOCint}` definiert den Pagestyle der auch für dieses Dokument (mit Ausnahme der vorangegangenen und der nächsten Seite) verwendet wurde (siehe auch Environment `DOC` in Abschnitt 2.1). Für zweiseitigen Druck:

Left page header:

Kapitel-Seite	Kapitelnummer Überschrift
---------------	---------------------------

Left page footer:

XRF-Revision	XRF-Titel
--------------	-----------

Right page header:

Unterkapitelnummer Überschrift	Kapitel-Seite
--------------------------------	---------------

Right page footer:

XRF-Titel	XRF-Revision
-----------	--------------

Wird einseitig gedruckt (wie bei diesem Dokument), so erzeugt `\pagestyle{DOCint}`:

Page header:

Kapitelnummer Überschrift	Kapitel-Seite
---------------------------	---------------

Page footer:

XRF-Titel	XRF-Revision
-----------	--------------

Wird statt `\begin{DOC}{xrfid}{xrftitle}{xrfrev}...\end{DOC}` nur `\begin{document}...\end{document}` verwendet oder wird für `xrftitle` und `xrfrev` eine Leerdefinition angegeben, so erscheint – wie auf dieser Seite – lediglich das aktuelle Datum am Außenrand des Seitenfusses.

3.3.3 Auftragsprotokoll: `\pagestyle{AUFTRAG}`

Das Kommando `\pagestyle{AUFTRAG}` definiert einen pagestyle fürs Auftragsprotokoll:

Page header:

I N T E S

Auftrag

Datum

Seite: Nummer / MaxPages

Das Kommando `\Kopf{Auftrag}{Datum}` gibt hierfür Werte ein.

Am Ende des Textes muß das Kommando `\lastpage` folgen (definiert `\label{maxpag}`).

Example : The style of this page was defined via:

```
\pagestyle{AUFTRAG}
\Kopf{Auftrag}{July 1990}
```

3.4 Document Sectioning

3.4.1 Document Section: `\docsec[1]`

`\docsec{chapter}` defines a new document section with title chapter, increments the section counter, starts new right-hand page and resets section-pagecounter, -figurecounter and -tablecounter to 1. During the \TeX -Translation an additional comment for the processed section is typed on the screen.

4 General Utilities

4.1 Additional Symbols

Macro	Symbol	Description
<code>\3</code>	β	German "EsZ"
<code>\eq</code>	\triangleq	Equivalent to . . .
<code>\earrow</code>	\longrightarrow	East arrow (text arrow to the right)
<code>\warrow</code>	\longleftarrow	West arrow (text arrow to the left)
<code>\hdots</code>	\dots	Horizontal dots

All following symbols use modified `\stackrel` macro. The argument is set on top of the mathematic symbol. The mathematic symbol itself is set in *scriptstyle*.

Macro	Symbol	Description
<code>\EQ{1}</code>	$\stackrel{!}{=}$	Equals a number or variable
<code>\NE{a}</code>	$\stackrel{a}{\neq}$	Not equal a number or variable
<code>\LT{0}</code>	$\stackrel{0}{<}$	Less than a number or variable
<code>\LE{A}</code>	$\stackrel{A}{\leq}$	Less or equal a number or variable
<code>\GT{-3}</code>	$\stackrel{-3}{>}$	Greater than a number or variable
<code>\GE{?}</code>	$\stackrel{?}{\geq}$	Greater or equal a number or variable

4.2 Emphasized Text for Headings, Titles or Remarks

4.2.1 Paragraph Header: `\head[1]`

Insert a `\goodbreak` and print a boldface paragraph header with extra vertical space to the preceding and following text.

Usage : `\head{text}`

Example : This is the presceding text `\head{This is bold text`
(usually for headlines)} and this is the following text.

This is the presceding text

This is bold text (usually for headlines)

and this is the following text.

4.2.2 Underlined Paragraph Header: `\uhead[1]`

Insert a `\goodbreak` and print an underlined paragraph header with extra vertical space to the preceeding and following text.

Usage : `\uhead{text}`

Example : This is the presceeding text `\uhead{This is a underlined text (usually for headlines)}` and this is the following text.

This is the presceeding text

This is a underlined text (usually for headlines)

and this is the following text.

4.2.3 Paragraph Header (Big and Bold): `\bigbf[1]`

Insert a `\goodbreak` and print a large and boldface paragraph header with extra vertical space [2mm] to the preceeding and following text.

Usage : `\bigbf{text}`

Example : This is the presceeding text `\bigbf{This is large and bold (usually for headlines)}` and this is the following text.

This is the presceeding text

This is large and bold (usually for headlines)

and this is the following text.

4.2.4 Dynamic Titeling: `\titel...\TITEL`

Various titles from paragraph up to front page titeling, each one centered and preceeded by a `\goodbreak`:

Example : `\titel{Text in a \/\titel\{\hdots\} macro !}`
`\Titel{Text in a \/\Titel\{\hdots\} macro !}`
`\TITel{Text in a \/\TITel\{\hdots\} macro !}`
`\TITeL{Text in a \/\TITeL\{\hdots\} macro !}`
`\TITEL{Text in a \/\TITEL\{\hdots\} macro !}`
`\TITEL{Text in a \/\TITEL\{\hdots\} macro !}`

Text in a `\titel{...}` macro !

Text in a `\Titel{...}` macro !

Text in a `\TITel{...}` macro !

Text in a `\TITeL{...}` macro !

Text in a `\TITEL{...}` macro !

Text in a `\TITEL{...}` macro !

4.3 Note and Boxes

4.3.1 Note with Top and Bottom Line: `\note[1]`

`\note` produces an indented paragraph in italic letters between two horizontal rules.

Usage : `\note{text}`

Example : `\note{This is an important note !}`

This is an important note !

4.3.2 Text Box: `\Kasten[1]`

`\Kasten` produces an indented text box. The box must be used with a new paragraph.

Usage : `\Kasten{text}`

Example : `\Kasten{Text \\ second line...}`

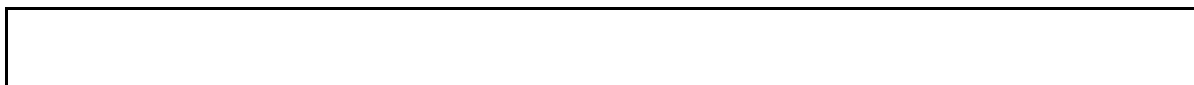
Text
second line...

4.3.3 Empty Box: `\Rahmen[1]`

`\Rahmen` produces an empty box in an extra paragraph. The box uses the whole `\textwidth`.

Usage : `\Rahmen{height}`
 height : Box height (unit is not optional)

Example : `\Rahmen{10mm}`



4.4 Extensions to the `\verbatim` Environment

4.4.1 Include File as ASCII-Text: `\verbatimfile`[1]

Usage : `\verbatimfile{filename}`
Includes the file as a normal `\verbatim`-output.

Example : `\verbatimfile{intex.text}`

```
This is an ASCII text for the \verbatim example
in the "LaTeX at INTES" documentation.
This is the 3rd line.
This is the 4th line.
You can use \verbatimfile or \verbatimlisting for
FORTRAN source code for instance !
This is the last line.
```

4.4.2 Include File as Listing: `\verbatimlisting`[1]

Usage : `\verbatimlisting{filename}`
Includes the file as a `\verbatim`-output with automatic line-numbering on the left side (like FORTRAN listings).

Example : `\verbatimlisting{intex.text}`

Table with Automatic Page Breaking			- 1 -
			16.3.2023
Parameter	Type	Short description	
1 ALTRED	Logical	Alternate stiffness/load reduction	
2 APPC	BCD	SOL200 (STATICS,MODES,BUCK)	
3 AUTOSPC	Logical	Automatic generation of SPCs	
4 AUTOSPCR	Logical	GPSP AUTOSPC, SOL66	
5 BAILOUT	Integer	GPSP MAXRATIO,stop or continue near singularity	
6 BETA	Real	Factor for transient heat transfer, NEWMARK	
7 BIGER	Real	SUPER,S1, OUT element stress → BIGER	
8 BIGER1	Real	SUPER,S1, OUT eq. BIGER for composite elements	
9 BIGER2	Real	SUPER,S1, OUT eq. BIGER for comp. el. failure indices	
10 BUCKLE	Integer	For nonlinear buckling analysis in restarts	
11 CB1	Real	Factor for damping matrix of superelements	
12 CB2	Real	Factor for damping matrix of superelements	
13 CHECKOUT	Logical	if YES in SOL same results as SOL66 (Superelements)	
14 CK1	Real	Factor for stiffness matrix of superelements o. GENEL	
15 CK2	Real	Factor for stiffness matrix of superelements o. GENEL	
16 CK3	Real	Factor for stiffness matrix of superelements m. GENEL	
17 CLOSE	Real	SCRSPEC user param	
18 CM1	Real	Factor for mass matrix of superelements	
19 CM2	Real	Factor for mass matrix of superelements	
20 CNSTRT	Integer	Control of restarts SOL66	
21 COMPARE	Integer	SOL66 only	
22 COUPMASS	Integer	> 0 Generation of coupled masses. < 0 lumped masses	
23 CP1	Real	Factor for load vectors	
24 CP2	Real	Factor for load vectors	
25 CURV	Integer	CURV OUT stress/strain of 2-dim in MAT-CID	
26 CURVPLOT	Integer	CURVPLOT OUT all at GRID-CID	
27 DBCCONV	BCD	DBC POST Format	
28 DBCDIAG	Integer	DBC POST Prints error messages of DBC module	
29 DBCLEAN	Integer	DBDIR restart use old DB or not	
30 DBCOVWRT	Logical	DBC POST Overwrite FORTRAN units or not	
31 DBDICT	Integer	DBDIR OUT DB directory in DBDIR module	
32 DBDROPT	Integer	DBDIR OUT Ddata blocks and param	
33 DBDRVER	Integer	DBDIR select version	
34 DBDRPRJ	Integer	DBDIR select project	
35 DBDRNL	Integer	DBDIR OUT also nonlinear DB directory	
36 DBINIT	Integer	DBDIR use of DB blocks all/dynamic init	
37 DBNAME	BCD	DBDIR name of DB	
38 DBNBLKS	Integer	DBDIR max number of DB blocks, mach. dep.	
39 DBSET1	Integer	subset DB	
40 DBSET2	Integer	subset DB	
41 DBSET3	Integer		
42 DBSET4	Integer		
43 DBSORT	Integer	DBDIR sorts DB directory printout	
44 DDRMM	Integer	DDRMM	
** Additional explanation of used symbols			

<div> <div>Table with Automatic Page Breaking</div> <div>- 1 -</div> </div>			
16.3.2023			
Parameter	Type	Short description	
45 DLOAD	Integer	CURVPLOT controls x-spacing effect of differential stiffness Matrices of Design SA min. absolute value of D. SA suppress the print	
46 DOPT	Integer		
47 DSNOKD	Real		
48 DSZERO	Real		
** Additional explanation of used symbols			

4.5 Short Commands `\beg`, `\B` and `\E` for Environments

Instead of the normal `\begin{name} ... \end{name}` environments one can use the abbreviated `\beg{name} ... \end{name}` or even shorter: `\B{name} ... \E{name}`. This improves the source-readability of deeply nested environments (i.e. when describing algorithms like in section 7), e.g.:

<code>\beg{name}</code>		<code>\begin{name}</code>
<code>\B{name}</code>		<code>\begin{name}</code>
<code>\B{name}</code>		<code>\begin{name}</code>
<code>\B{name}</code>	instead of	<code>\begin{name}</code>
<code>\E{name}</code>		<code>\end{name}</code>
<code>\E{name}</code>		<code>\end{name}</code>
<code>\E{name}</code>		<code>\end{name}</code>
<code>\end{name}</code>		<code>\end{name}</code>

4.5.1 Dirty Tricks with `\begin{BEcheckOFF}...\end{BEcheckOFF}`

If you are using very deep environment nesting (e.g. more than 10 levels on describing algorithms) you will run against the \TeX -Limit *TeX capacity exceeded, sorry [main memory size 65536]* and the \TeX run will be terminated. One can enlarge this limit when using the `\begin{BEcheckOFF}...\end{BEcheckOFF}` environment in combination with the abbreviations `\B` and `\E` from above. `BEcheckOFF` will switch off the \LaTeX environment stack for all `\B... \E` subenvironments (the handling of normal `\begin... \end` environments will stay unchanged) – but remember `BEcheckOFF` is a dirty trick, because \LaTeX can't check the correct stacking of corresponding `\B... \E` statements (e.g. closing a `\B{If}` by `\E{Loop}` will cause no error). See section 7.6 on pages 7-11 for an example

4.6 More Utilities

4.6.1 Actual Date: `\heute`

Example : This document was interpreted and printed
at `\heute` (day.month.year).

This document was interpreted and printed at 16.3.2023 (day.month.year).

4.6.2 Shortened Italic and Underline: `\I[1]` and `\U[1]`

`\I{text}` is an abbreviation for `{\it text}` (italic command with argument).

`\U{text}` is an abbreviation for `\underline{text}` (shortened macro), see example in 5.6.

4.6.3 Absolut Value: `\abs[1]`

`\abs{variable}` prints *variable* with mathematical absolute-quotes:

e.g.: `SUM=\abs{HUGO}+SUM` produces: `SUM=|HUGO|+SUM`

4.6.4 Dynamic Horizontal Spacing: `\leer...LEER`

Horizontal space which depends on the actual script size:

Macro <code>\quad</code> leaves this	empty space (using <code>\normalsize</code>).
Macro <code>\leer</code> leaves this	empty space (using <code>\normalsize</code>).
Macro <code>\Leer</code> leaves this	empty space (using <code>\normalsize</code>)..
Macro <code>\LEer</code> leaves this	empty space (using <code>\normalsize</code>).
Macro <code>\LEEr</code> leaves this	empty space (using <code>\normalsize</code>).
Macro <code>\LEER</code> leaves this	empty space (using <code>\normalsize</code>).

4.6.5 Bold List item: `\bfitem[1]`

`\bfitem{text}` prints a bold face list item (can be used just like `\item{text}`).

`\bfitem{text} \hat{=} \item[{\bf text}]`

See also `\bfItem` and `\Item` on chapter 10 (*Index Making*).

4.7 Environments for Listing Definitions

See also ARGLIST and ARGDES environments on pages 5-2ff (sections 5.2 and 5.3) !

4.7.1 Left Adjusted Items: `\B{deflist}...\E{deflist}`

The `deflist` environment uses the \LaTeX `list` environment. The `\labelwidth` is set with the environment argument. The label is set in boldface to the left.

```
\B{deflist}{This place reserves the label place}
  \item[All labels are set to the left]      Short descriptions needs more
                                             than one line, if a large
                                             {\tt\labelwidth} is used.

  \item[The {\tt\labelwidth} ist constant] But the place for the
                                             description is small.

  \item[Short labels]                       have not more place.
\E{deflist}
```

All labels are set to the left	Short descriptions needs more than one line, if a large <code>\labelwidth</code> is used.
The <code>\labelwidth</code> ist constant	But the place for the description is small.
Short labels	have not more place.

4.7.2 Right Adjusted Items: `\B{listdef}...\E{listdef}`

The `listdef` environment uses the \LaTeX `list` environment. The `\labelwidth` is set with the environment argument. The label is set in boldface to the right.

```
\B{listdef}{This place reserves the label place}
  \item[All labels are set to the left]      Short descriptions needs more
                                             than one line, if a large
                                             {\tt\labelwidth} is used.

  \item[The {\tt\labelwidth} ist constant] But the place for the
                                             description is small.

  \item[Short labels]                       have not more place.
\E{listdef}
```

All labels are set to the left	Short descriptions needs more than one line, if a large <code>\labelwidth</code> is used.
The <code>\labelwidth</code> ist constant	But the place for the description is small.
Short labels	have not more place.

5 Describing Functions and Subroutines

5.1 Definition of Function/Subroutine: `\B{Call}[3]...\E{Call}`

Usage	<pre> : \B{Call}{title}{name}{arguments} % 3 arguments \B{arg1} \item[Synopsis] % What is the functionality of this subroutine ? \item[Input] % Description of all input arguments (if any). \item[Transient] % Description of transient arguments (if any). \item[Output] % Description of output arguments (if any). \item[Errors] % List of error conditions. \item[Example] % Verbatim example calling \texttt{name} : : \E{arg1} \E{Call} </pre>		
Items	:	title	:
		name	: Name of the function/subroutine. It is printed in boldface, defined as a primary index entry and inserted in the list-of-calls (with arguments).
		arguments	: The arguments are optional. They are printed in <code>\small</code> between brackets.

5.2 List of Arguments: `\B{arg1...}\End{arg1...}`

The ARGLIST environment uses the L^AT_EX list environment. The `\labelwidth` is set with the environment argument. The label is set to the left. Between the items is an additional `\itemsep`.

The `argl` and `arglist` environments uses the ARGLIST environment with a fixed `\labelwidth`.

The ARGLIST list:

```
\B{ARGLIST}{XX}
  \item[A] The {\tt\labelwidth} can choose in the {\tt ARGLIST} environment
            with the argument. It is an additional
  \item[B] {\tt\itemsep} between the items.
\E{ARGLIST}
```

A : The `\labelwidth` can choose in the ARGLIST environment with the argument. It is an additional

B : `\itemsep` between the items.

The argl list:

```
\B{argl}
  \item[Input] The {\tt argl} environment uses the {\tt ARGLIST} environment
               with a small {\tt\labelwidth}.
  \item[Output] It is an additional {\tt\itemsep} between the items.
\E{argl}
```

Input : The `argl` environment uses the ARGLIST environment with a small `\labelwidth`.

Output : It is an additional `\itemsep` between the items.

The arglist list:

```
\B{arglist}
  \item[Input] The {\tt arglist} environment uses the {\tt ARGLIST} environment
               with a large {\tt\labelwidth}.
  \item[Output] It is an additional {\tt\itemsep} between the items.
\E{arglist}
```

Input : The `arglist` environment uses the ARGLIST environment with a large `\labelwidth`.

Output : It is an additional `\itemsep` between the items.

5.3 Description of Arguments: $\backslash B\{argd...\}...\backslash E\{argd...\}$

The ARGDES environment uses the L^AT_EX list environment. It is used for the description of arguments in a list environment. The $\backslash labelwidth$ is set with the environment argument. The label is set to the left.

The label of the $arg\$$ environment is set in mathematic mode. The $arg\$$, $argd$, $argdes$ and $argdescription$ environments uses the ARGDES environment with a fix $\backslash labelwidth$.

The ARGDES descriptionlist:

```
\B{ARGDES}{XXXX}
  \item[A] The {\tt\labelwidth} can choose in the {\tt ARGDES} environment
           with the argument. It is no additional
  \item[B] {\tt\itemsep} between the items.
\E{ARGDES}
```

A : The $\backslash labelwidth$ can choose in the ARGDES environment with the argument. It is no additional
 B : $\backslash itemsep$ between the items.

The $arg\$$ descriptionlist:

```
\B{arg$}
  \item[=0] The {\tt arg\$} environment uses the {\tt ARGDES} environment
           with a mathematic label and a very small {\tt\labelwidth}.
  \item[>0] It is no additional {\tt\itemsep} between the items.
\E{arg$}
```

= 0 : The $arg\$$ environment uses the ARGDES environment with a mathematic label and a very small $\backslash labelwidth$.
 > 0 : It is no additional $\backslash itemsep$ between the items.

The $argd$ descriptionlist:

```
\B{argd}
  \item[in] The {\tt argd} environment uses the {\tt ARGDES} environment
           with a small {\tt\labelwidth}.
  \item[out] It is no additional {\tt\itemsep} between the items.
\E{argd}
```

in : The $argd$ environment uses the ARGDES environment with a small $\backslash labelwidth$.
 out : It is no additional $\backslash itemsep$ between the items.

The argdes descriptionlist:

```

\B{argdes}
  \item[input]   The {\tt argdes} environment uses the {\tt ARGDES}
                  environment with a large {\tt\labelwidth}.
  \item[output]  It is no additional {\tt\itemsep} between the items.
\E{argdes}

```

input : The argdes environment uses the ARGDES environment with a large \labelwidth.
output : It is no additional \itemsep between the items.

The argdescription descriptionlist:

```

\B{argdescription}
  \item[inputargument] The {\tt argdescription} environment uses the
                        {\tt ARGDES} environment with a very large
                        {\tt\labelwidth}.
  \item[outputargument] It is no additional {\tt\itemsep} between the items.
\E{argdescription}

```

inputargument : The argdescription environment uses the ARGDES environment with a very large \labelwidth.
outputargument : It is no additional \itemsep between the items.

5.4 Argument Item with Array Counter: $\backslash\text{ARGi}[2]$

Synopsis : Item command for the description of array elements. $\backslash\text{ARGi}$ is intended to be used within an argument description (e.g. `argd` list) of an `Call` respectively `call` environment.

Usage : $\backslash\text{ARGi}\{\text{arrayindex}\}\{\text{argument}\}$ % 2 arguments

Items : `arrayindex` : Index of argument within listed array.
`argument` : Name of argument to be described. The standard `labelwidth` for `argument` depends on the list environment used.

Example : $\backslash\text{B}\{\text{argd}\}$
 $\backslash\text{item}[\text{NOFROW}]$ Integer; Row dimension of Level-2 matrix.
 $\backslash\text{item}[\text{NOFCOL}]$ Integer; Column dimension of Level-2 matrix.
 $\backslash\text{item}[\text{IWIN}]$ Integer(4); Current window limits of non-zero Level-2 data:
 $\backslash\text{B}\{\text{argd}\}$
 $\backslash\text{ARGi}\{1\}\{\text{MINROW}\}$ Index of first non-zero row (N/W corner of data window).
 $\backslash\text{ARGi}\{2\}\{\text{MINCOL}\}$ Index of first non-zero column (N/W corner of data window).
 $\backslash\text{ARGi}\{3\}\{\text{MAXROW}\}$ Index of last non-zero row (S/E corner of data window).
 $\backslash\text{ARGi}\{4\}\{\text{MAXCOL}\}$ Index of last non-zero column (S/E corner of data window).
 $\backslash\text{ARGi}\{99\}\{\text{Dummy}\}$ Extra item (just to see the adjusting of large indices).
 $\backslash\text{E}\{\text{argd}\}$
 $\backslash\text{E}\{\text{argd}\}$

`NOFROW` : Integer; Row dimension of Level-2 matrix.

`NOFCOL` : Integer; Column dimension of Level-2 matrix.

`IWIN` : Integer(4); Current window limits of non-zero Level-2 data:

- (1) `MINROW` : Index of first non-zero row (N/W corner of data window).
- (2) `MINCOL` : Index of first non-zero column (N/W corner of data window).
- (3) `MAXROW` : Index of last non-zero row (S/E corner of data window).
- (4) `MAXCOL` : Index of last non-zero column (S/E corner of data window).
- (99) `Dummy` : Extra item (just to see the adjusting of large indices).

5.5 List of Errors: `\B{errlist}...\E{errlist}`

Synopsis : List environment for the description of PERMAS error conditions. `errlist` is intended to be stacked within the top level `argl` list of an `Call` respectively `call` environment.

Usage : `\B{errlist}` % No arguments
`\item[X]` Description of X-Error % Each item describes a condition
`\item[F]` : : F-Error % which (may) lead to an PERMAS
`\item[E]` : : E-Error % error of the error severity
`\item[W]` : : W-Error % indicated by the item argument
: : : :
`\E{errlist}`

Example : `\B{argl}`
`\item[Errors]`
`\B{errlist}`
`\item[X]` If I3REG is not bound to any Level-3 matrix.
`\item[X]` If the pointer value of I3REG is undefined,
`\` i.e. if I3REG is not assigned to a valid Level-3 matrix.
`\item[F]` If the regarding Level-2 matrix does not exist
`\` i.e. I3REG points to a nil Level-2 matrix (no input).
`\item[E]` Description of an Error condition.
`\item[W]` Description of a condition for Warnings.
`\E{errlist}`
`\E{argl}`

Errors : **X** If I3REG is not bound to any Level-3 matrix.
X If the pointer value of I3REG is undefined,
i.e. if I3REG is not assigned to a valid Level-3 matrix.
F If the regarding Level-2 matrix does not exist
i.e. I3REG points to a nil Level-2 matrix (no input).
E Description of an Error condition.
W Description of a condition for Warnings.

5.6 Summary Example

```

\B{call}{This example uses the complete list environments:}
    {HUGO}{INPUT1, INPUT2, \I{TRANSIENT}, \U{OUTPUT}}
\B{argl}
  \item[Synopsis]   What is the functionality of this subroutine ?
  \item[Input]      Description of all input arguments.
                    \B{argd}
                    \item[{\tt INPUT1}] Integer; First input argument.
                    \item[{\tt INPUT2}] Integer; Second input argument.
                    \E{argd}
  \item[Transient]  \B{argdes}
                    \item[\I{TRANSIENT}] Logical; Transient variable e.g. OK.
                    \E{argdes}
  \item[Output]     Description of all output arguments.
                    \B{argd}
                    \item[\U{OUTPUT}] Real; Output argument.
                    \E{argd}
  \item[Errors]     \B{errlist}
                    \item[X] If I3REG is not bound to any Level-3 matrix.
                    \item[F] If the regarding Level-2 matrix does not exist.
                    \item[E] Description of an Error condition.
                    \item[W] Description of a condition for Warnings.
                    \E{errlist}
\E{argl}
\E{call}

```

This example uses the complete list environments:

HUGO (INPUT1, INPUT2, *TRANSIENT*, OUTPUT)

Synopsis : What is the functionality of this subroutine ?

Input : Description of all input arguments.

INPUT1 : Integer; First input argument.

INPUT2 : Integer; Second input argument.

Transient : *TRANSIENT* : Logical; Transient variable e.g. OK.

Output : Description of all output arguments.

OUTPUT : Real; Output argument.

Errors : *X* If I3REG is not bound to any Level-3 matrix.

F If the regarding Level-2 matrix does not exist.

E Description of an Error condition.

W Description of a condition for Warnings.

6 Describing Data Structures

6.1 Fortran Vector: \B{DMSvector}...\E{DMSvector}

Usage : \B{DMSvector}
 \DMShead{type}{name}{description} % Optional
 \DMScolsep{factor} % Optional
 \addrow{item1}{description1}
 \addrow{item2}{description2}
 :
 \Addrow{posi}{itemi}{descriptioni}
 :
 \Seprow{subtitle}{description}
 :
 \E{DMSvector}

Items : \DMShead : Optional specification of vector header (type and/or name and/or description can be empty):
 type : Data type (left-upper subtitle)
 name : Vector-Name (left-upper title), will be defined as a primary help & index entry.
 description : Short description of name.
 \DMScolsep : Factor for the horizontal space between vertical lines and column names, default is 1.0 (varifies width of table).
 \addrow : Adds a new row with automatic row numbering, item name itemi and a short description of item element.
 \Addrow : Adds a new row with explicitly given position name posi, item name itemi and a short description of item element.

Example 1 : \B{DMSvector}
 \DMShead{Integer}{HUGO}{Vector of Hided and Unidentified Grafical Objects}
 \addrow{IVALUE}{Description of IVALUE}
 \addrow{NXTVAL}{Next value in array's column}
 \addrow{NEWROW}{New row in array}
 \cline{2-2}
 \addrow{ISEP}{Extra item separated from preceeding entries}
 \Addrow{ i }{MORE}{Item entry with given row position}
 \Addrow{max}{LAST}{Last item in this array}
 \E{DMSvector}

<i>Integer</i>	HUGO	<i>Vector of Hided and Unidentified Grafical Objects</i>
1	IVALUE	Description of IVALUE
2	NXTVAL	Next value in array's column
3	NEWROW	New row in array
4	ISEP	Extra item separated from preceeding entries
i	MORE	Item entry with given row position
max	LAST	Last item in this array

6.2 Fortran Array: \B{DMSarray}[2]...\E{DMSarray}

Usage : \B{DMSarray}{icol}{nofcol} % 2 arguments
 \DMShead{type}{name}{description} % Optional
 \DMScolsep{factor} % Optional
 \addrow{item1}{description1}
 \addrow{item2}{description2}
 : : :
 \Addrow{posi}{itemi}{descriptioni}
 : : : :
 \Seprow{subtitle}{description}
 : : :
 \E{DMSarray}

Items : **icol** : Column index, counts the columns (may be empty).
nofcol : Maximal column index (may be empty).
\DMShead : Optional specification of array header (type and/or name and/or description can be empty):
 type : Data type (left-upper subtitle)
 name : Array-Name (left-upper title), will be defined as a primary help & index entry.
 description : Short description of name.
\DMScolsep : Factor for the horizontal space between vertical lines and column names, default is 1.0 (varifies width of table).
\addrow : Adds a new row with automatic row numbering, item name itemi and a short description of item element.
\Addrow : Adds a new row with explicitly given position name posi, item name itemi and a short description of item element.
\Seprow : Separate rows by an extra subtitle.

Example 1 : \B{DMSarray}{ICol}{MaxCol}
 \DMShead{Integer}{HUGO}{Array of Hided and Unidentified Grafical Objects}
 \addrow{IVALUE}{Description of IVALUE}
 \addrow{NXTVAL}{Next value in array's column}
 \addrow{NEWROW}{New row in array}
 \Addrow{ i }{MORE}{Item entry with given row position}
 \Addrow{max}{LAST}{Last item in this array}
 \E{DMSarray}

HUGO: *Array of Hided and Unidentified Grafical Objects*

Integer

	1	...	ICol	...	MaxCol	
1	IVALUE					Description of IVALUE
2	NXTVAL					Next value in array's column
3	NEWROW					New row in array
i	MORE					Item entry with given row position
max	LAST					Last item in this array

Example 1 : \B{DMSarray}{}{NOFCOL}
 \DMShead{Real}{HUGO}{Just an example with no column counter}
 \addrow{RVAL}{Description of RVAL}
 \addrow{RNXT}{Next value in array's column}
 \Addrow{max}{LAST}{Last item in this array}
 \E{DMSarray}

HUGO: *Just an example with no column counter*

Real

	1	...	<i>i</i>	...	NOFCOL	
1	RVAL				Description of RVAL
2	RNXT				Next value in array's column
max	LAST					Last item in this array

Example 2 : \B{DMSarray}{ICOL}{}
 \DMShead{Real}{HUGO}{Just an example with no maxima}
 \addrow{RVAL}{Description of RVAL}
 \addrow{RNXT}{Next value in array's column}
 \Addrow{max}{LAST}{Last item in this array}
 \E{DMSarray}

HUGO: *Just an example with no maxima*

Real

	1	...	ICOL	...	max	
1	RVAL				Description of RVAL
2	RNXT				Next value in array's column
max	LAST					Last item in this array

6.3 DMS Table: \B{DMStable}[2]...\E{DMStable}

Usage : \B{DMStable}{ituple}{noftup} % 2 arguments
 \DMShead{type}{name}{description} % Optional
 \DMScounter{countername} % Optional
 \DMScolsep{factor} % Optional
 \DMSform{short} % Optional
 \addcol{columnindex1}{columnname1}
 \addcol{columnindex2}{columnname2}
 : : : : :
 \addtuple{rowindex1}
 \addvdots
 \addtuple{rowindex2}
 \addcline
 \E{DMStable}

Items : ituple : Tuple index, counts the tuple (can be empty).
 noftup : Maximal tuple index, an empty specification {} will result in a DMS table with open bottom.
 \DMShead : Optional specification of table header (type and/or name and/or description can be empty):
 type : Data type (left-upper subtitle)
 name : DMS-Name (left-upper title), will be defined as a primary help & index entry.
 description : Short description of name
 \DMScounter : The extra countername is used as an header for the column of tuple indices, default is to have no counter name.
 \DMScolsep : Factor for the horizontal space between vertical lines and column names, default is 1.0 (varifies width of table).
 \DMSform : Vertical form (short or long), default is \DMSform{long}
 \addcol : Adds a new column at the right side of the DMS table, places columnindex over and columnname under the top line of that column.
 \addvdots : Optionally append a line with vertical dots.
 \addtuple : Optionally append a line with given rowindex.
 \addcline : Optionally append an additional separation line.

```
Example 1 : \B{DMStable}{Jcount}{Maxtup}
           \DMShead{Integer}{HUGO}{HUngrY G0ose}
           \addcol{1}{First} \addcol{2}{Second}
           \addcol{3}{Third} \addcol{i}{. . .}
           \addcol{N}{Last}
           \E{DMStable}
```

HUGO: *HUngrY G0ose*

<i>Integer</i>	1	2	3	i	N
	First	Second	Third	...	Last
1					
2					
3					
⋮					
Jcount					
⋮					
Maxtup					

```
Example 2 : \B{DMStable}{IOLD}{MAXOLD}
           \DMShead{Integer}{IDUM}{Pointer}
           \DMScounter{ITUPLE=}
           \addcol{1}{INEW}
           \E{DMStable} \hspace*{-20ex}
           \B{DMStable}{INEW}{MAXNEW}
           \DMShead{Real}{HUGO}{Compacted table}
           \DMSform{short}
           \addcol{1}{First}
           \addcol{2}{Second}
           \addcol{3}{Third}
           \E{DMStable}
```

IDUM: *Pointer*

<i>Integer</i>	1
ITUPLE=	INEW
1	
2	
3	
⋮	
IOLD	
⋮	
MAXOLD	

HUGO: *Compacted table*

<i>Real</i>	1	2	3
	First	Second	Third
1			
⋮			
INEW			
⋮			
MAXNEW			

```
Example 3 : \B{DMStable}{Index}{}  
           \addcol{1}{First}  
           \addcol{2}{Second}  
           \addcol{3}{Third}  
           \addcol{i}{Hugo-A}  
           \addcol{i+1}{Hugo-B}  
           \addcol{i+2}{Hugo-C}  
           \addcol{\hdots}{Hugo-\hdots}  
           \E{DMStable}
```

	1	2	3	i	i+1	i+2	...
	First	Second	Third	Hugo-A	Hugo-B	Hugo-C	Hugo-...
1							
2							
3							
⋮							
Index							
⋮							

```
Example 4 : \B{DMStable}{}{Maxtup}  
           \DMShead{Integer}{HUGO}  
           {This short description is wider than the DMStable}  
           \addcol{1}{First}  
           \addcol{2}{Second}  
           \addcol{3}{Third}  
           \addcol{i}{. . .}  
           \addcol{N}{Last}  
           \E{DMStable}
```

HUGO: *This short description is wider than the DMStable*

<i>Integer</i>	1	2	3	i	N
	First	Second	Third	...	Last
1					
2					
3					
⋮					
Maxtup					

```
Example 5 : \B{DMStable}{Jcount}{}  
           \DMShead{Integer}{HUGO}  
           {Here the width of the DMStable is stretched by DMScolsep}  
           \DMScounter{\UL{ICOUNT}}  
           \DMScolsep{3}  
           \addcol{1}{First}  
           \addcol{2}{Second}  
           \addcol{3}{Third}  
           \addcol{i}{. . .}  
           \addcol{N}{Last}  
           %  
           \addcline  
           \addtuple{ISTART}  
           \addvdots  
           \addtuple{ISTART+NX-1}  
           \addcline  
           \addvdots  
           \addtuple{NofTup}  
           \addcline  
           \E{DMStable}
```

HUGO: *Here the width of the DMStable is stretched by DMScolsep*

<i>Integer</i>	1	2	3	i	N
<u>ICOUNT</u>	First	Second	Third	. . .	Last
1					
2					
3					
⋮					
Jcount					
⋮	⋮	⋮	⋮	⋮	⋮
ISTART					
⋮	⋮	⋮	⋮	⋮	⋮
ISTART+NX-1					
⋮	⋮	⋮	⋮	⋮	⋮
NofTup					

6.4 DMS Control Table: \B{DMSCtable}...\E{DMSCtable}

```

Usage      : \B{DMSCtable}                % No arguments
           \DMShead{type}{name}{description} % Optional
           \DMScounter{countername}          % Optional
           \DMScolsep{factor}                % Optional
           \addcol{columnindex1}{columnname1}
           \addcol{columnindex2}{columnname2}
           :           :           :           :           :
           \newtup T(1,1) & T(1,2) & T(1,3) ...
           \newtup T(2,1) & T(2,2) & T(2,3) ...
           :           :           :           :           :
           \Newtup{tupi} T(i,1) & T(i,2) & T(i,3) ...
           :           :           :           :           :
           \E{DMSCtable}

```

Items	:	\DMShead	:	Optional specification of table header
		\DMShead	:	Optional specification of table header (type and/or name and/or description can be empty):
		type	:	Data type (left-upper subtitle)
		name	:	DMS-Name (left-upper title), will be defined as a primary help & index entry.
		description	:	Short description of name
		\DMScounter	:	The extra countername is used as an header for the column of tuple indices, default is to have no counter name.
		\DMScolsep	:	Factor for the horizontal space between vertical lines and column names, default is 1.0 (varifies width).
		\addcol	:	Adds a new column at the right side of the DMS table, places columnindex over and columnname under the topline of that column.
		\newtup	:	Start a new tuple definition with automatic tuple numbering and with each tuple element seperated by an ampersand (&).
		\Newtup	:	\newtup alternative: Start a new tuple definition without numbering, but with tuple name tupi (can be empty) and with each tuple element seperated by an ampersand (&).
		T(i, j)	:	Name of a tuple element in the i.th row and j.th column of the DMS control table.

Example 1 : `\B{DMSCTable}`
`\DMShead{Integer}{HUGO}{Control table}`
`\addcol{1}{First}`
`\addcol{2}{Second}`
`\addcol{3}{Third}`
`\addcol{i}{. . .}`
`\addcol{N}{Last}`
`\newtup Hugo11 & Hugo12 & Hugo13 & Hugo1i & Hugo1X`
`\newtup Hugo21 & Hugo22 & Hugo23 & Hugo2i & Hugo2X`
`\newtup Bertai & Bertai2 & Bertai3 & Bertai & bertaiX`
`\E{DMSCTable}`

HUGO: *Control table*

<i>Integer</i>	1	2	3	i	N
	First	Second	Third	. . .	Last
1	Hugo11	Hugo12	Hugo13	Hugo1i	Hugo1X
2	Hugo21	Hugo22	Hugo23	Hugo2i	Hugo2X
3	Bertai	Bertai2	Bertai3	Bertai	bertaiX

Example 2 : `\B{DMSCTable}`
`\renewcommand{\arraystretch}{2.0}`
`\DMShead{Integer}{HUGO}{Control table (using arraystretch)}`
`\DMScounter{Counter}`
`\DMScolsep{3.0}`
`\addcol{1}{First}`
`\addcol{2}{Second}`
`\addcol{3}{Third}`
`\addcol{i}{. . .}`
`\addcol{N}{Last}`
`\Newtup{1} Hugo11 & Hugo12 & Hugo13 & Hugo1i & Hugo1X`
`\Newtup{\vdots} Hugo21 & Hugo22 & Hugo23 & Hugo2i & Hugo2X`
`\Newtup{MAX} Bertai & Bertai2 & Bertai3 & Bertai & bertaiX`
`\E{DMSCTable}`

HUGO: *Control table (using arraystretch)*

<i>Integer</i>	1	2	3	i	N
Counter	First	Second	Third	. . .	Last
1	Hugo11	Hugo12	Hugo13	Hugo1i	Hugo1X
\vdots	Hugo21	Hugo22	Hugo23	Hugo2i	Hugo2X
MAX	Bertai	Bertai2	Bertai3	Bertai	bertaiX

6.5 DMS Sequence: \B{DMSsequence}[2]...\E{DMSsequence}

Usage : \B{DMSsequence}{ilist}{noflist} % 2 arguments
 \DMShead{type}{name}{description} % Optional
 \DMScounter{countername} % Optional
 \DMScolsep{factor} % Optional
 \DMSform{short} % Optional
 \addcol{columnindex1}{columnname1}
 \addcol{columnindex2}{columnname2}
 : : : : :
 \E{DMSsequence}

Items : ilist : List index, counts the lists in sequence (can be empty).
 noflist : Maximal list index, an empty specification {} will result in an DMS sequence with open bottom.
 \DMShead : Optional specification of sequence header (type and/or name and/or description can be empty):
 type : Data type (left-upper subtitle)
 name : DMS-Name (left-upper title), will be defined as a primary help & index entry.
 description : Short description of name
 \DMScounter : The extra countername is used as an header for the column of list indices, default is to have no counter name.
 \DMScolsep : Factor for the horizontal space between vertical lines and column names, default is 1.0 (varifies width).
 \DMSform : Vertical form (short or long), default is \DMSform{long}
 \addcol : Adds a new column at the right side of the DMS sequence, places columnindex over and columnname under the topline of that column.

Example 1 : \B{DMSsequence}{ILIST}{NOFLST}
 \DMShead{Integer}{HUGO}{Sequence}
 \addcol{1}{First}
 \addcol{2}{Second}
 \addcol{3}{Third}
 \addcol{i}{. . .}
 \addcol{N}{Last}
 \E{DMSsequence}

HUGO: Sequence

<i>Integer</i>	1	2	3	i	N
	First	Second	Third	...	Last
1					
ILIST	:	:	:	:	...
NOFLST					

```
Example 2      : \B{DMSsequence}{ILIST}{NOFLST}
                \DMShead{Integer}{HUGO}{Sequence}
                \DMScounter{mytext}
                \DMSform{short}
                \addcol{1}{First}
                \addcol{2}{Second}
                \addcol{3}{Third}
                \addcol{i}{. . .}
                \E{DMSsequence}
```

HUGO: *Sequence*

<i>Integer</i>	1	2	3	i
mytext	First	Second	Third	...
ILIST	:	:	:	...
NOFLST				

```
Example 3      : \B{DMSsequence}{List}{}
                \addcol{1}{First}
                \addcol{2}{Second}
                \addcol{3}{Third}
                \addcol{i}{Hugo-A}
                \addcol{i+1}{Hugo-B}
                \addcol{i+2}{Hugo-C}
                \addcol{\hdots}{Hugo-\hdots}
                \E{DMSsequence}
```

	1	2	3	i	i+1	i+2	...
	First	Second	Third	Hugo-A	Hugo-B	Hugo-C	Hugo-...
1							
List	:	:	:	:	:	:	...

6.6 DMS Control Sequence: $\backslash B\{DMSCsequence\}...\backslash E\{DMSCsequence\}$

Usage : $\backslash B\{DMSCsequence\}$ % No arguments
 $\backslash DMShead\{type\}\{name\}\{description\}$ % Optional
 $\backslash DMScounter\{countername\}$ % Optional
 $\backslash DMScolsep\{factor\}$ % Optional
 $\backslash addcol\{columnindex1\}\{columnname1\}$
 $\backslash addcol\{columnindex2\}\{columnname2\}$
: : : : :
 $\backslash newlist\{list1\} L(1,1) \& L(1,2) \& L(1,3) \dots$
 $\backslash newlist\{list2\} L(2,1) \& L(2,2) \& L(2,3) \dots$
: : : : :
 $\backslash Newlist\{listi\} L(i,1) \& L(i,2) \& L(i,3) \dots$
: : : : :
 $\backslash E\{DMSCsequence\}$

Items : $\backslash DMShead$: Optional specification of sequence header (type and/or name and/or description can be empty):
type : Data type (left-upper subtitle)
name : DMS-Name (left-upper title), will be defined as a primary help & index entry.
description : Short description of name
 $\backslash DMScounter$: The extra countername is used as an header for the column of list indices, default is to have no counter name.
 $\backslash DMScolsep$: Factor for the horizontal space between vertical lines and column names, default is 1.0 (varifies width).
 $\backslash addcol$: Adds a new column at the right side of the DMS sequence, places columnindex over and columnname under the topline of that column.
 $\backslash newlist$: Start a new list definition with automatic list numbering, list name listi (can be empty) and with each list element seperated by an ampersand (&).
 $\backslash Newlist$: $\backslash newlist$ alternative: Start a new list definition without numbering, but with list name listi (can be empty) and with each list element seperated by an ampersand (&).
 $L(i,j)$: Name of a list element in the i.th list and j.th column of the DMS control sequence.

Example 1 : `\B{DMSCsequence}`
`\DMShead{Integer}{HUGO}{Control sequence}`
`\DMScounter{Id \&\ Name}`
`\addcol{1}{First}`
`\addcol{2}{Second}`
`\addcol{i}{. . .}`
`\addcol{N}{Last}`
`\newlist{Liste} Hugo11 & Hugo12 & Hugo1i & Hugo1X`
`\newlist{LISTE} Hugo21 & Hugo22 & Hugo2i`
`\newlist{MORE} Berta1 & Berta2 & Bertai & BertaX`
`\E{DMSCsequence}`

HUGO: *Control sequence*

<i>Integer</i>	1	2	i	N
Id & Name	First	Second	. . .	Last
1 Liste	Hugo11	Hugo12	Hugo1i	Hugo1X
2 LISTE	Hugo21	Hugo22	Hugo2i	
3 MORE	Berta1	Berta2	Bertai	BertaX

Example 2 : `\B{DMSCsequence}`
`\renewcommand{\arraystretch}{1.5}`
`\DMShead{Integer}{HUGO}{Control sequence}`
`\DMScolsep{2.0}`
`\addcol{1}{First}`
`\addcol{2}{Second}`
`\addcol{i}{. . .}`
`\addcol{N}{Last}`
`\Newlist{1} Hugo11 & Hugo12 & Hugo1i & Hugo1X`
`\Newlist{\vdots} Hugo21 & Hugo22 & Hugo2i`
`\Newlist{MAX} Berta1 & Berta2 & Bertai & BertaX`
`\E{DMSCsequence}`

HUGO: *Control sequence*

<i>Integer</i>	1	2	i	N
	First	Second	. . .	Last
1	Hugo11	Hugo12	Hugo1i	Hugo1X
⋮	Hugo21	Hugo22	Hugo2i	
MAX	Berta1	Berta2	Bertai	BertaX

6.7 DMS List: \B{DMSlist}[1]...\E{DMSlist}

Usage : \B{DMSlist}{indexname} % 1 argument
 \DMShead{type}{name}{description} % Optional
 \DMScolsep{factor} % Optional
 \addcol{columnindex1}{columnname1}
 \addcol{columnindex2}{columnname2}
 : : : : :
 \E{DMSlist}

Items : indexname : List index and/or list name, can be empty.
 \DMShead : Optional specification of list header (type and/or name
 and/or description can be empty):
 type : Data type (left-upper subtitle)
 name : DMS-Name (left-upper title), will be
 defined as a primary help & index entry.
 description : Short description of name
 \DMScolsep : Factor for the horizontal space between vertical lines and column names,
 default is 1.0 (varifies width).
 \addcol : Adds a new column at the right side of the DMS list, places columnindex
 over and columnname under the topline of that column.

Example 1 : \B{DMSlist}{LISTNAM}
 \DMShead{Integer}{HUGO}{Sequence name of following list}
 \addcol{1}{First}
 \addcol{2}{Second}
 \addcol{i}{Hugo-A}
 \addcol{i+1}{Hugo-B}
 \addcol{i+2}{Hugo-C}
 \addcol{\hdots}{Hugo-\hdots}
 \E{DMSlist}

HUGO: *Sequence name of following list*

<i>Integer</i>	1	2	i	i+1	i+2	...
LISTNAM	First	Second	Hugo-A	Hugo-B	Hugo-C	Hugo-...

Example 2 : `\B{DMSlist}{\bf{LONG}}`
`\DMShead{Integer}{}{}`
`\DMScolsep{3}`
`\addcol{1}{First}`
`\addcol{2}{Second}`
`\addcol{3}{Third}`
`\addcol{i}{. . .}`
`\addcol{N}{Last}`
`\E{DMSlist}`

<i>Integer</i>	1	2	3	i	N
LONG	First	Second	Third	. . .	Last

Example 3 : `\B{DMSlist}{\bf{SHORT}}`
`\DMShead{Integer}{}{}`
`\DMScolsep{0.5}`
`\addcol{1}{First}`
`\addcol{2}{Second}`
`\addcol{3}{Third}`
`\addcol{i}{. . .}`
`\addcol{N}{Last}`
`\E{DMSlist}`

<i>Integer</i>	1	2	3	i	N
SHORT	First	Second	Third	. . .	Last

6.8 DMS Control List: \B{DMSClist}...\End{DMSClist}

Usage : \B{DMSClist} % No arguments
 \DMShead{type}{NAME}{description} % Optional
 \DMScolsep{factor} % Optional
 \addlin{elementname1}{short description 1}
 \addlin{elementname2}{short description 2}
 : : : : :
 \Addlin{index1}{elementname1}{short description 1}
 \Addlin{index2}{elementname2}{short description 2}
 : : : : :
 \E{DMSClist}

Items : \DMShead : Optional specification of list header (type and/or name and/or description can be empty):
 type : Data type (left-upper subtitle)
 name : DMS-Name (left-upper title), will be defined as a primary help & index entry.
 description : Short description of name
 \DMScolsep : Factor for the horizontal space between vertical lines and column names, default is 1.0 (varifies width).
 \addlin : Adds a new line to the DMS control list: Name of list element and short description.
 \Addlin : \addlin alternative: Adds a new line to the DMS control list: Arbitrary element index, name of list element and short description.

Example 1 : \B{DMSClist}
 \DMShead{Integer}{HUGO}{Control list}
 \addlin{NAME1}{Description1}
 \addlin{NAME2}{Description2}
 \addlin{ i }{ . . . }
 \addlin{LAST}{Blabla}
 \E{DMSClist}

Integer	HUGO	Control list
1	NAME1	Description1
2	NAME2	Description2
3	i	. . .
4	LAST	Blabla

Example 2 : \B{DMSClist}
 \addlin{i}{Hugo-A}
 \addlin{i+1}{Hugo-B}
 \addlin{i+2}{Hugo-C}
 \E{DMSClist}

1	i	Hugo-A
2	i+1	Hugo-B
3	i+2	Hugo-C

Example 3 : \B{DMSClist}
 \DMShead{Integer}{HUGO}{Control list}
 \Addlin{Index1}{NAME1}{Description1}
 \Addlin{Index2}{NAME2}{Description2}
 \Addlin{\vdots}{ i }{ . . . }
 \Addlin{MAX}{LAST}{Blabla}
 \E{DMSClist}

<i>Integer</i>	HUGO	<i>Control list</i>
Index1	NAME1	Description1
Index2	NAME2	Description2
\vdots	i	...
MAX	LAST	Blabla

6.9 Naming Rule for List Names: `\namlis[3]`

`\namlis` specifies a naming rule for data lists (description or example for list names) out of 3 A4-Integer. This macro uses the `argdes` environment.

Usage : `\namlis{namlis1}{namlis2}{namlis3}` % DMS list name

Items : `namlis1` : Integer value of NAMLIS(1)
 `namlis2` : Integer value of NAMLIS(2)
 `namlis3` : Integer value of NAMLIS(3)

Example : `\namlis{namlis1}{namlis2}{namlis3}` % DMS list name

Name of lists : NAMLIS (1) : `namlis1`
 NAMLIS (2) : `namlis2`
 NAMLIS (3) : `namlis3`

6.10 Title for Textual Description of Data Objects: `\DMStitle[3]`

Usage : `\DMStitle{name}{type}{description}`

Items : `name` : Name of data object (will be defined as an index entry).
 `type` : Data type and/or object type (e.g. "Integer", "Sequence", "Real", "Book" or "Double Precision Scratch Table")
 `description` : Text description of name.

This macro uses the `argd` environment. For alternative DMS-titeling see also macro `\DMShead` in the environments of subsections 6.3 to 6.8.

Example : `\DMStitle{HUGO}{Integer \ DMS-Table}{Text for description \hdots}`

HUGO : *Integer DMS-Table*
 Text for description . . .

6.11 DMS Book: $\backslash\text{B}\{\text{DMSbook}\}[4]\dots\backslash\text{E}\{\text{DMSbook}\}$

Usage : $\backslash\text{B}\{\text{DMSbook}\}\{\text{irow}\}\{\text{nrow}\}\{\text{icol}\}\{\text{ncol}\}$ % 4 arguments
 $\backslash\text{DMShead}\{\text{type}\}\{\text{name}\}\{\text{description}\}$ % Optional
 $\backslash\text{DMSform}\{\text{matrixformat}\}$ % Optional
 $\backslash\text{E}\{\text{DMStable}\}$

Not yet implemented !

6.12 Local Redefinition of DMStype for Help-Files: $\backslash\text{DMStype}[1]$

Usage : $\backslash\text{DMStype}\{\text{objecttype}\}$ % DNS Type of Data Object for .hlp file

Items : objecttype : DMS type: Array, Table, Sequence, List or Book.

7 Describing Algorithms

7.1 Module Environment: `\begin{Module}[2]...\end{Module}`

Usage : `\def\Modulesep{factor}` % Optional
 `\setlength{\Modulesep}{dimension}` % Optional
 `\begin{Module}{title}{abstract}` % 2 arguments
 Substructures and text for
 description of algorithm % See next
 : : : % sections
 `\end{Module}`

The Module environment produces a primary index of title (if not empty). Alternatively there is a module environment which does the same as Module except no index entry will be defined.

Items : `\Modulesep` : Defining the module separating factor (0.0...0.2). This specifies the horizontal space between the vertical outlines and the included text in relation to `\Modulewidth`. Specification is optional (default value is 0.02).

 `\Modulewidth` : Width for the following module and Module environments. Specification is optional (default value is 1.0\textwidth).

 title : Module title (symbolic name or name of implemented routine), can be empty. A non-empty title in the Module environment will be defined as a primary index.

 abstract : Short description of functional aim, can be empty.

Example 1 : `\begin{module}{Symbolic name}{Example with no operation}`
 `\begin{Block}`
 Text and substructures like blocks,
 loops and case structures. \\
 The Block environment used in this example
 will be described in the next section.
 `\end{Block}`
 `\end{module}`

Symbolic name

Example with no operation

Text and substructures like blocks, loops and case structures.
The Block environment used in this example will be described in the next section.

Example 2 : `\def\Modulesep{0.0}`
 `\setlength{\Modulewidth}{50mm}`
 `\begin{Module}{HUGO}{}`
 `\begin{Block}`
 This example has an empty `{\tt abstract}`
 and leaves no horizontal space between
 the outlines and the included text.
 `\end{Block}`
 `\end{Module}`

HUGO

This example has an empty abstract and leaves no horizontal space between the outlines and the included text.

Example 3 : `\def\Modulesep{0.2}`
 `\setlength{\Modulewidth}{0.6\textwidth}`
 `\begin{module}{}{Example with empty title definition}`
 `\begin{Block}`
 This example leaves a lot of horizontal space
 between the outlines and the included text.
 `\end{Block}`
 `\end{module}`

Example with empty title definition

This example leaves a lot of horizontal space between the outlines and the included text.

Example 4 : `\fboxrule2pt`
 `\begin{module}{}{}`
 `\begin{Block}`
 This example has an empty title and an empty
 abstract definition. `\` It uses the standard
 `{\tt \Modulesep}` and `{\tt \Modulewidth}`. `\`
 The thickness of the lines is defined via the
 normal `\LaTeX`-dimension `{\tt \fboxrule}`.
 `\end{Block}`
 `\end{module}`

This example has an empty title and an empty abstract definition. It uses the standard `\Modulesep` and `\Modulewidth`. The thickness of the lines is defined via the normal `\LaTeX`-dimension `\fboxrule`.

7.2 Separating Blocks: `\begin{Block}...\end{Block}` and `\block[1]`

Usage : Within Module respectively module environments:

```

:      :
\begin{Block}                                % No arguments
      Substructures and text for
      description of this block              % See next
      :      :      :                      % sections
\end{Block}
:      :                                % Shortform for
\block{text}                                % pur text blocks
:      :                                % with 1 argument

```

Example : `\setlength{\Modulewidth}{0.6\textwidth}`
`\begin{module}{HUGO}{Example with 4 separate blocks}`
`\begin{Block}` First block with or without
substructures `\end{Block}`
`\begin{Block}`
Second block which may contain
substructures like loops or
case structures (those may contain
further imbedded `{\tt Block}`
environments or `{\tt \block}` macros.
`\end{Block}`
`\block{Single block without substructure (text only)}`
`\begin{Block}` Last block with substructers `\end{Block}`
`\end{module}`

HUGO

<i>Example with 4 separate blocks</i>

First block with or without substructures

Second block which may contain substructures like loops or case structures (those may contain further imbedded Block environments or <code>\block</code> macros.
--

Single block without substructure (text only)

Last block with substructers

7.3 Definition of Loops: `\begin{Loop}[1]...\end{Loop}` and `\loopblock[2]`

Usage : Within Module respectively module environments:

```

:      :
\def\Modulesep{factor}           % Optional
:      :
\begin{Loop}{loopcondition}      % 1 argument
    Substructures and text for
    description of this loop      % See other
:      :      :                  % sections
\end{Loop}
:      :
\loopblock{loopcondition}{text}  % Shortform for
:      :                        % pur text blocks
:      :                        % with 2 arguments

```

Items : `\Modulesep` : The module separating factor described in section 7.1 also specifies the horizontal insert of every new loop level (the left insert is $2*factor*\Modulewidth$).

`loopcondition`: Text describing the loop parameters (may be empty).

`text` : Pur text description without any further substructures.

Example 1 : `\setlength{\Modulewidth}{0.7\textwidth}`
`\begin{module}{HUGO}{Example with varifying Module separator}`
`\begin{Loop}{For all elements}`
`\loopblock{x times}{Standard {\tt\Modulesep}}`
`\end{Loop}`
`\def\Modulesep{0.1}`
`\begin{Loop}{For all elements}`
`\loopblock{x times}{Enlarged {\tt\Modulesep}}`
`\end{Loop}`
`\end{module}`

HUGO

Example with varifying Module separator

Loop: For all elements

Loop: x times

Standard `\Modulesep`

Loop: For all elements

Loop: x times

Enlarged `\Modulesep`


```

Example 2 : \def\Modulesep{0.05}
           \setlength{\Modulewidth}{0.6\textwidth}
           \begin{module}{\NoName algorithm with four loops}
             \block{Preceding blocks or substructures}
             \begin{Loop}{For all elements}
               \begin{Block}
                 Text and substructures like blocks,
                 loops and case structures.
               \end{Block}
             \end{Loop}
             \block{Intermediate blocks or substructures}
             \begin{Loop}{ $N=1,2,\dots,LIMIT$ }
               \loopblock{5 times}{Loop within a loop}
               \loopblock{}{Loop with empty {\tt loopcondition}}
             \end{Loop}
             \block{Additional blocks or substructures}
           \end{module}

```

<i>NoName algorithm with four loops</i>	
Preceding blocks or substructures	
Loop: For all elements	
	Text and substructures like blocks, loops and case structures.
Intermediate blocks or substructures	
Loop: $N = 1, 2, \dots, LIMIT$	
	Loop: 5 times
	Loop within a loop
	Loop:
	Loop with empty loopcondition
Additional blocks or substructures	

7.4 Case Structures: `\begin{If}...\end{If}` and `\ifblock[2]` `\begin{Case}...\end{Case}` and `\caseblock[2]`

Usage : Within Module respectively module environments:

```

:      :
\def\Modulesep{factor}           % Optional
\settowidth{\casewidth}{mask}   % Optional
:      :
\begin{If}{ifcondition}          % 1 argument
  \begin{Case}{casecondition}    % 1 argument
    Substructures and text for
    description of this case     % See other
    :      :      :             % sections
  \end{Case}
  :      :                       % Shortform for
  \caseblock{casecondition}{text} % pur text blocks
  :      :                       % with 2 arguments
\end{If}
:      :                       % Shortform for
\ifblock{ifcondition}{text}      % pur text blocks
:      :                       % with 2 arguments

```

Items : `\Modulesep` : The module separating factor described in section 7.1 also specifies the horizontal insert of every new if level.

`\casewidth` : Width for the following `casecondition` boxes. (the default value is set to the standard mask width of XXXXX).

`ifcondition` : Text describing the if condition (may be empty).

`casecondition` : Text describing the case condition (may be empty).

`text` : Pur text description without any further substructures.

Example 1 : `\setlength{\Modulewidth}{0.8\textwidth}`
`\begin{module}{HUGO}{Example with varifying Module separator}`
`\begin{If}{ $X=0$?}`
`\begin{Case}{Yes}`
`\ifblock{Normal}{Using standard {\tt\}/Modulesep}}`
`\end{Case}`
`\caseblock{No}{Single case block without substructures}`
`\end{If}`
`\def\Modulesep{0.1}`
`\begin{If}{ $Y=?$ }`
`\begin{Case}{ $=0$ }`
`\ifblock{Special}{Using enlarged {\tt\}/Modulesep}}`
`\end{Case}`
`\caseblock{ $=1$ }{Single case block without substructures}`
`\caseblock{others}{Perform an error message}`
`\caseblock{}{Empty {\tt casecondition}`
`(just for demonstration)}`
`\end{If}`
`\end{module}`

HUGO

<i>Example with varifying Module separator</i>
--

If: $X = 0 ?$

Yes	<table border="1"> <tr> <td>If: Normal</td> </tr> <tr> <td>Using standard <code>\Modulesep</code></td> </tr> </table>	If: Normal	Using standard <code>\Modulesep</code>
If: Normal			
Using standard <code>\Modulesep</code>			

No	Single case block without substructures
----	---

If: $Y = ?$

$= 0$	<table border="1"> <tr> <td>If: Special</td> </tr> <tr> <td>Using enlarged <code>\Modulesep</code></td> </tr> </table>	If: Special	Using enlarged <code>\Modulesep</code>
If: Special			
Using enlarged <code>\Modulesep</code>			

$= 1$	Single case block without substructures
-------	---

others	Perform an error message
--------	--------------------------

Empty casecondition (just for demonstration)
--

Example 2 : \setlength{\Modulewidth}{0.7\textwidth}
 \begin{module}{HUGO}{Example with varifying Case width}
 \begin{If}{ $X=0$?}
 \begin{Case}{Yes}
 \ifblock{Normal}{Using standard {\tt\casewidth}}
 \end{Case}
 \caseblock{No}{Single case block without substructures}
 \end{If}
 \settowidth{\casewidth}{XXXXXXXXXX}
 \begin{If}{ $Y=?$ }
 \begin{Case}{ $=0$ }
 \ifblock{Special}{Using enlarged {\tt\casewidth}}
 \end{Case}
 \caseblock{ $=1$ }{Single case block without substructures}
 \caseblock{others}{Perform an error message}
 \caseblock{}{Empty {\tt casecondition}
 (just for demonstration)}
 \end{If}
 \end{module}

HUGO

<i>Example with varifying Case width</i>
--

If: $X = 0$?	
Yes	If: Normal
	Using standard \casewidth
No	Single case block without substructures

If: $Y = ?$	
$= 0$	If: Special
	Using enlarged \casewidth
$= 1$	Single case block without substructures
others	Perform an error message
Empty casecondition (just for demonstration)	

Example 3 : Using the abbreviations `\B` and `\E` from section 4.5:

```
\setlength{\Modulewidth}{0.8\textwidth}
\B{module}{HUGO}{Algorithm with stacked Case Structure}
  \block{Preceding block without substructure}
  \B{If}{ $X=?$ }
    \B{Case}{ $\$=0\$$ }
      \B{If}{ $\$Y=0\$ \ ?$ }
        \caseblock{Yes}{Case block without substructure}
        \B{Case}{No} \block{Doing something}
        \ifblock{More ?}{Text and substructures \hdots}
      \E{Case}
    \E{If}
  \E{Case}
  \B{Case}{ $\$=1\$$ } \block{This can be done by a
    {\tt \caseblock} macro also.} \E{Case}
  \B{Case}{ $\$=2\$$ }
    \loopblock{For all values}{Single loop block in case}
  \E{Case}
  \caseblock{ $\$>2\$$ }{Do something (nothing for all other cases)}
  \E{If} \block{Following blocks}
\E{module}
```

HUGO

<i>Algorithm with stacked Case Structure</i>
--

Preceding block without substructure

If: $X = ?$

If: $Y = 0 ?$

Yes

Case block without substructure

$= 0$

No

Doing something

If: More ?

Text and substructures ...

$= 1$

This can be done by a <code>\caseblock</code> macro also.

$= 2$

Loop: For all values

Single loop block in case

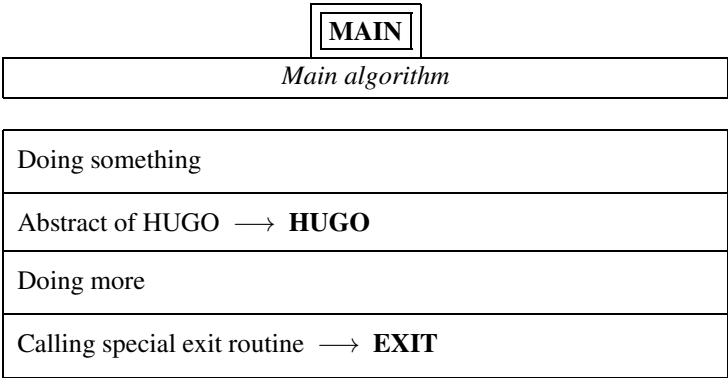
> 2

Do something (nothing for all other cases)
--

Following blocks

7.5 Referring to Submodules: `\call[1]` and `\Call[1]`

Usage	: \call{title}	or	\Call{title}	% 1 argument
Item	: title	: Title of referred submodule (symbolic name or name of implemented routine). On the \Call version the title will be defined as an index entry additionally.		
Example	<pre>: \setlength{\Modulewidth}{0.6\textwidth} \begin{module}{MAIN}{Main algorithm} \block{Doing something} \block{Abstract of HUGO \Call{HUGO}} \block{Doing more} \block{Calling special exit routine \call{EXIT}} \end{module}</pre>			



7.6 Summary Example

With the abbreviations `\B` and `\E` from section 4.5 this example uses the `\begin{BEcheckOFF}...\end{BEcheckOFF}` environment to switch off the LaTeX environment stack. This allows a deeper nesting of loops and case structures. Otherwise this example would run against *TeX capacity exceeded, sorry [main memory size 65536]* and terminate the interpretation – but remember `BEcheckOFF` is a dirty trick, because LaTeX can't check the correct stacking of corresponding `\B... \E` statements.

```
\begin{BEcheckOFF}
\def\Modulesep{0.05}
\B{module}{MAIN}{Abstract of this algorithm}
  \block{This is text describing the first block}
  \block{Now it may be necessary to call a submodule \call{HUGO}}
  \B{Loop}{This is a very large loop}
    \block{This is the first block in loop}
    \B{Loop}{This is a smaller loop}
      \loopblock{This is a loop in a loop in a loop}
        {Description of the loop}
      \loopblock{This is a loop direct following the upper loop}
        {Description of the loop}
    \E{Loop}
  \block{This is a block between two loops}
  \B{Loop}{One more loop}
    \B{If}{Test condition (no operation if condition is false)}
      \B{Case}{YES case}
        \ifblock{subcondition}{Single block in lonely case}
        \B{If}{}
          \caseblock{$X=1$}
            {First block of a case structure without
             a preceding {\tt\caseblock} statement.}
          \B{Case}{$X=2$}
            \block{Doing something}
            \loopblock{This is a loop within a case}
              {Deepest stacked block}
          \E{Case}
          \caseblock{$X=3$}{Last case, do nothing for other X.}
        \E{If}
      \block{Text and substructures like blocks,
             loops and case structures.}
    \E{Case}
  \E{If}
\E{Loop}
\block{Last block}
\E{Loop}
\E{module}
\end{BEcheckOFF}
```

Last block

8 Describing Syntax Structures

8.1 Syntax Environment: `\begin{Syntax}[3]...\end{Syntax}`

```
Usage      : \SboxBCtrue           % Default is \SboxBCfalse
            \setlength{\Sboxrule}{dimension} % Optional
            \setlength{\Sboxwidth}{dimension} % Optional
            \begin{Syntax}{title}{abstract}{keyword} % 3 arguments
              Submacros and text describing
              the syntax of keyword                % See next
              :      :      :                      % sections
            \end{Syntax}
```

The Syntax environment produces a primary index and a \TeX label definition of keyword. Alternatively there is a syntax environment which does the same as Syntax except no index entry or label will be defined.

Both create a syntax box including a `displaymath`-like environment with bold face keyword as entry. Besides the standard \TeX / \LaTeX -math-macros all macros of the following pages may be used to enlighten the syntax description.

```
Items      : \SboxBCtrue   : Invokes big syntax boxes with description vertical and horizontal centered
                                within.
            \SboxBCfalse  : Invokes standard sized syntax boxes with vertical centered and left-adjusted
                                description within within (default).
            \Sboxrule     : Defines the line width used to draw the syntax box independently of normal
                                \fboxrule (default is 0.1ex).
            \Sboxwidth    : Width of syntax box (default value is 0.9\textwidth).
            title         : Title written ahead of the syntax box (in large size and boldface), may be
                                empty.
            abstract      : Short summary description of keyword, may be empty.
            keyword       : Keyword to be described, will be used as main entry (first item in syntax
                                box).
```

```
Example 1   : \begin{syntax}{Example with no description}{}{HUGO}\end{syntax}
```


9 Bibliographic References

```
\begin{docbibliography}{muster_marke}  
  Eintragungen  
\end{docbibliography}
```

Waiting for description !

10 Index Making

The section 10.1.1 is an abstract of the L^AT_EX documentation:

MakeIndex: An Index Processor For L^AT_EX

from *Leslie Lamport*. The complete description can be found on the Apollo under:

//node_cd21/arbortext/index/doc/makeindex.tex

More about the use of *MakeIndex* in appendix E.

10.1 Definition of Index Entries

10.1.1 The Basics: \index[1]

Each \index command causes L^AT_EX to write an entry on the idx file. The following example shows some simple \index commands and the index entries that they produce. The page number refers to the page containing the text where the \index command appears.

Page ii:	\index{Alpha}	Alpha, ii
Page viii:	\index{alpha}	alpha, viii, ix, 22
Page ix:	\index{alpha}	alpha bet, 24
	\index{Alphabet}	Alphabet, ix
Page 22:	\index{alpha}	alphabet, 23
	\index{alphas}	alphas, 22
Page 23:	\index{alphabet}	
	\index{alphabet}	
Page 24:	\index{alpha bet}	

Note that the duplicate \index{alphabet} commands on page 23 produce only one “23” in the index.

Subentries

To produce a subentry, the argument of the \index command should contain both the main entry and the subentry, separated by a ! character.

Page 7:	\index{gnat!size of}	gnat, 32
Page 32:	\index{gnat}	anatomy, 35
Page 35:	\index{gnat!anatomy}	size of, 7
	\index{gnus!good}	gnus
Page 38:	\index{gnus!bad}	bad, 38
		good, 35

Alphabetical Position

If you specify an entry of the form `abc@text`, the string `abc` determines the alphabetical position of the entry, while the string `text` produces the text of the entry. This feature is useful because the argument of the `\index` command provides the actual input string that \LaTeX uses to generate the index entry. In the following example, the input `\alpha` produces the symbol α ,

Page 12:	<code>\index{alphas}</code>		alpha, 13
Page 13:	<code>\index{alpha}</code>		α , 14
Page 14:	<code>\index{alpha@\$\alpha\$}</code>		alphas, 12

10.1.2 Primary Index: `\indexbf[1]`

`\indexbf{text}` writes a primary index entry with a boldface page number as primary reference, see the HUGO examples in the index.

10.1.3 Print and Create an Index: `\Index[1]`

Print the input text and create an index entry of it:

<code>\Index{text}</code>	: does the same as <code>text\index{text}</code>
<code>\Indexbf{text}</code>	: does the same as <code>text\indexbf{text}</code>
<code>\bfIndex{text}</code>	: does the same as <code>{\bf text}\index{text}</code>
<code>\bfIndexbf{text}</code>	: does the same as <code>{\bf text}\indexbf{text}</code>
<code>\keyword{text}</code>	: In opposite to the <code>\Index</code> macro <code>\keyword</code> produces an additional blank between the printed word and the following text (e.g. <code>\keyword{X}-2</code> prints X -2, <code>\Index{X}-2</code> produces X-2).

10.1.4 List Item and Create an Index: `\Item[1]`

Use the input text as an list-`\item`, print it and create an index entry of it:

<code>\Item{text}</code>	: does the same as <code>\item[text]\index{text}</code>
<code>\Itembf{text}</code>	: does the same as <code>\item[text]\indexbf{text}</code>
<code>\bfItem{text}</code>	: does the same as <code>\item[{\bf text}]\index{text}</code>
<code>\bfItembf{text}</code>	: does the same as <code>\item[{\bf text}]\indexbf{text}</code>

10.2 Index Layout

If You wish to create Your own index layout You may use the `THEINDEX` environment from the `intes.sty` file as a master example. `THEINDEX` produces an index layout in double column format, with each paragraph a separate entry. The user commands `\item`, `\subitem` and `\subsubitem` are used to produce the entries, and `\indexspace` adds an extra vertical space to put above the first entry with a new letter of the alphabet.

The `THEINDEX` environment is used if You enclose a `\printdocindex` or a `\DOCindex` in Your main document, see section 2.1 on page 2-1 and the index of this document for example.

A Calling TeX: `intex[g] myfile[.tex] [onlyfile]`

Start the \LaTeX file translation with the following System command:

```
latex myfile[.tex] [onlyfile]
or
intex myfile[.tex] [onlyfile]
Action: myfile.tex (and myfile.aux/ind/lof/lot/toc/xrf) →
        myfile.dvi (and myfile.aux/idx/lof/lot/toc/xrf)
```

- `latex myfile[.tex] [onlyfile]`
The translation with the `latex` command needs the `intes` style file in the `\documentstyle[...]` brackets.
- `intex myfile[.tex] [onlyfile]`
The translation with the `intex` commands works similar as the `latex` translation. The `intes` style is included with the startup of the `intex` translation. The `intes` style must not be included with the `\documentstyle[...]`.

B Output to Screen: preview myfile[.dvi]

Start the PREVIEWer (DVI server for screen) with the following System command:

```
preview myfile[.dvi]
```

(Action: myfile.dvi → Screen of Your Workstation)

Type ? (cursor inside the active PREVIEW window) and you will get the following list:

ArborText Preview Help

Commands:

a	- display information about what is under the cursor
c	- center the viewing area around the cursor
e	- go to the last page
h	- move viewing area to default home position
H	- move viewing area to absolute home position
ctrl-H	- set default home position at cursor and move there
i	- invert video (black-on-white or white-on-black)
o	- superimpose a page border outline
q	- quit and return to Unix
wp	- display pages in "page" mode, one at a time
ws	- display pages in "scroll" mode
wt	- display pages in "twoup" mode, two at a time
/, \	- scan forward (backward) for text scan string
?	- display this information

These commands can take a numeric repetition factor as a prefix:

b	- go backward one or more pages
ctrl-D, d, D	- move viewing area down in small, medium, big steps
f	- go forward one or more pages
ctrl-L, l, L	- move viewing area left in small, medium, big steps
m, M	- magnify this page (all pages) in magsteps
ctrl-R, r, R	- move viewing area right in small, medium, big steps
s, S	- shrink this page (all pages) in magsteps
ctrl-U, u, U	- move viewing area up in small, medium, big steps

The page command can take a ordinal page number as a numeric prefix. If the numeric prefix contains a decimal point, the page command uses it as an absolute page number. For example, p means go to the first page, 12p goes to the 12th page, and 42.p goes to page 42.

p	- go to first (or specified) page
---	-----------------------------------

C Creating Output for HP-Laser: dvihp myfile[.dvi]

Start the DVILASER/HP (DVI server for HP Laser Jet) with the following command:

```
dvihp myfile[.dvi]
```

(Action: myfile.dvi and plots.pcl → myfile.hp)

C.1 DVILASER Options

Type ? (after the DVILASER option> prompt) and you will get the following list:

DVILASER/HP Apollo version 4.4.0

Copyright (c) 1987 by ArborText, Inc. All rights reserved.

TeX output 1990.05.21:1619

DVILASER option> ?

DVILASER options:	Example
-----	-----
number <num-pages>	num 100
width <width>	width 8.5in
height <height>	height 11in
shift <amount>	shift .5in
xy <xoffset> <yoffset>	xy 72pt 2.54cm
cycle <number>	cycle 1
erase	era
copies <num-copies>	cop 1
magnification <num>	mag 1000
reverse [on off]	rev
datemark [on off []] [<text>]	datemark on 0 "Page Revised 1/1/86"
printer [LaserJet+ LJ+] [cart=<char>]	printer LJ+ cart=F
fontlist	fontlist
fontsub <file>	fontsub mysubfile
overlay on off [<file>]	overlay on myoverlayfile
@file	@myoptfile
pixelpath [<path> [<path> ...]]	pi /arbortext/pixels/canon300
tfmpath [<path> [<path> ...]]	t /arbortext/fonts
portrait	port
landscape	land
interactive	i
verbose	ver
options	o
reset	res
scaling [on off]	scaling
quit	q
?	?

C.2 Interactive Options

With `i` (after the DVILASER option> prompt) you are entering the interactive mode. Herein – type `?` (after the `(./.) [.]`: prompt) and you will get the following list:

```
DVILASER option> i
(1/2) [1]: ?
```

```
Options available: (all can be abbreviated to one character)
yes                => yes, do this page
no                 => no, don't do this page
y 9                => yes, do next 9 pages
n10                => no, skip next 10 pages
n-3                => no, move back 3 pages
y -9               => yes, plus previous 8 pages
goto 3             => no, and go to absolute page 3
search 9           => no, and move forward to DVI page [9]
s                  => no, and repeat last search
reverse 1.2        => no, and move backward to DVI page [1.2]
downto * 22        => yes, down until DVI page [*.22]
upto <=33 * >44    => yes, up until DVI page [<=33.*.>44]
cmd <DVILASER command> => e.g., cmd xy 1in 1in
@file              => read page commands from file
exit               => all done, exit program
quit               => abort processing with error return
options            => prints current option settings
?                  => prints this command summary
```

D Printing HP-Laser Files: `laser myfile[.hp]`

Send your laser file to the printer with the following System command:

```
laser myfile[.hp]
```

(Action: `myfile.hp` → HP Laser Jet Printer)

In bad days you make some of mistakes and LASERed it.
The only correction now is to cancel the LASER-Spool file.

```
$ /usr/bin/lpstat -t          -> laser-nnn
$ /usr/bin/cancel laser-nnn
$
```

or if you have `/usr/bin` in your command search rule

```
$ lpstat -t          -> laser-nnn
$ cancel laser-nnn
$
```

Example

```
$ /usr/bin/lpstat -t
scheduler is running
system default destination: laser
device for laser: //node_8a68/sys/node_data/dev/sio1
laser accepting requests since May  3 20:41
printer laser now printing laser-404.  enabled since May  3 20:41
laser-404          ast          454187   May 25 17:50 on laser
$ /usr/bin/cancel laser-404
request "laser-404" cancelled
$
```


E makeindex myfile[.idx]

The section E.1 is an abstract of the \LaTeX documentation

MakeIndex: An Index Processor For \LaTeX

from *Leslie Lamport*. The complete description can be found on the Apollo under:

`//node_cd21/arbortext/index/doc/makeindex.tex`

E.1 How to Use *MakeIndex*

MakeIndex is a program for making an index in a document generated with \LaTeX . The first step in producing the index is to put the necessary `\index` commands in your document, as described in the next section. Here, I describe how to generate the index after the `\index` commands are in place.

Let's suppose that the root file of your document is `myfile.tex`. You must make the following changes to your document:

- Add the `makeidx` document-style option to the list of options in the `\documentstyle` command \longrightarrow obsolete since Rev.F. of `intes.sty`
- Put a `\makeindex` command in the preamble (between the `\documentstyle` and `\begin{document}` commands) \longrightarrow obsolete since Rev.G with the use of `\DOC...` environment.
- Put a `\printindex`, `\printdocindex` or `\DOCindex` command where you want the index to appear — usually at the end, right before the `\end{document}` command.

You then run \LaTeX on your entire document, causing it to generate the file `myfile.idx`, which I will call the `idx` file. Next, run the *MakeIndex* program by typing the following System command:

`makeindex myfile[.idx]`

(Action: `myfile.idx` \longrightarrow `myfile.ind`)

This produces the file `myfile.ind`, which I will call the `ind` file. If *MakeIndex* generated no error messages, you can now rerun \LaTeX on your document and the index will appear.

See section 2.1 on page 2-1 for an example of `makeidx`, `\makeindex` and `\DOCindex` !

Index

\cong , 4-1
 \longrightarrow , 4-1
 \longleftarrow , 4-1
 \dots , 4-1
 $=$, 4-1
 \neq , 4-1
 $<$, 4-1
 \leq , 4-1
 $>$, 4-1
 \geq , 4-1

\mathbb{B} , 4-1

A

\backslash abs, 4-8
 \backslash addcline, 6-5
 \backslash addcol, 6-5, 6-9, 6-11, 6-13, 6-15
 \backslash Addlin, 6-17
 \backslash addlin, 6-17
 \backslash Addrow, 6-1, 6-2
 \backslash addrow, 6-1, 6-2
 \backslash addtuple, 6-5
 \backslash addvdots, 6-5
 argd, 5-3, 5-7, 6-19
 ARGDES, 4-9, 5-3
 argdes, 5-3
 argdescription, 5-3
 \backslash ARGi, 5-5
 argl, 5-2, 5-6, 5-7
 ARGLIST, 4-9, 5-2
 arglist, 5-2
 Argument Counter, 5-5
 Array, 6-2
 AUFTRAG, 3-7

B

\backslash B, 4-7, 7-11
 \backslash B{argd}, 5-3, 5-7, 6-19
 \backslash B{ARGDES}, 4-9, 5-3
 \backslash B{argdes}, 5-3
 \backslash B{argdescription}, 5-3
 \backslash B{argl}, 5-2, 5-6, 5-7
 \backslash B{ARGLIST}, 4-9, 5-2
 \backslash B{arglist}, 5-2
 \backslash B{Call}, 5-1, 5-6, 5-7
 \backslash B{call}, 5-1, 5-7

\backslash B{deflist}, 4-9
 \backslash B{DMSarray}, 6-2
 \backslash B{DMSbook}, 6-20
 \backslash B{DMSClist}, 6-17
 \backslash B{DMSCsequence}, 6-13
 \backslash B{DMSCtable}, 6-9
 \backslash B{DMSlist}, 6-15
 \backslash B{DMSsequence}, 6-11
 \backslash B{DMStable}, 6-5
 \backslash B{DMSvector}, 6-1
 BEcheckOFF, 4-7, 7-11
 \backslash beg, 4-7
 \backslash begin, 4-7
 \backslash begin{argd}, 5-3, 5-7, 6-19
 \backslash begin{ARGDES}, 4-9, 5-3
 \backslash begin{argdes}, 5-3
 \backslash begin{argdescription}, 5-3
 \backslash begin{argl}, 5-2, 5-6, 5-7
 \backslash begin{ARGLIST}, 4-9, 5-2
 \backslash begin{arglist}, 5-2
 \backslash begin{BEcheckOFF}, 4-7, 7-11
 \backslash begin{Block}, 7-3
 \backslash begin{Call}, 5-1, 5-6, 5-7
 \backslash begin{call}, 5-1, 5-7
 \backslash begin{Case}, 7-6
 \backslash begin{deflist}, 4-9
 \backslash begin{DMSarray}, 6-2
 \backslash begin{DMSbook}, 6-20
 \backslash begin{DMSClist}, 6-17
 \backslash begin{DMSCsequence}, 6-13
 \backslash begin{DMSCtable}, 6-9
 \backslash begin{DMSlist}, 6-15
 \backslash begin{DMSsequence}, 6-11
 \backslash begin{DMStable}, 6-5
 \backslash begin{DMSvector}, 6-1
 \backslash begin{DOC}, 2-1, 2-2, 2-4
 \backslash begin{docbibliography}, 9-1
 \backslash begin{document}, 2-1
 \backslash begin{errlist}, 5-6
 \backslash begin{If}, 7-6
 \backslash begin{listdef}, 4-9
 \backslash begin{Loop}, 7-4
 \backslash begin{Module}, 7-1
 \backslash begin{module}, 7-1
 \backslash begin{Syntax}, 8-1
 \backslash begin{syntax}, 8-1
 \backslash B{errlist}, 5-6
 \backslash bfIndex, 10-2
 \backslash bfIndexbf, 10-2
 \backslash bfItem, 10-2

`\bfititem`, 4-8
`\bfititembf`, 10-2
 bibliography, 9-1
`\bigbf`, 4-2
`\B{listdef}`, 4-9
 block, 7-3
`\block`, 7-3

C

Call, 5-1, 5-6, 5-7
`\Call`, 7-10
`\call`, 7-10
 call, 5-1, 5-7
 Case, 7-6
 Case width, 7-6
`\caseblock`, 7-6
`\casewidth`, 7-6
 Control List, 6-17
 Control Sequence, 6-13
 Control Table, 6-9
 Cross references, 2-4

D

Data objects, 6-1
`deflist`, 4-9
`\DINA4`, 3-3
 Dirty tricks, 4-7
 DMS list, 6-19
 DMS objects, 6-1
 DMS type, 6-20
`DMSarray`, 6-2
`DMSbook`, 6-20
`DMSClist`, 6-17
`\DMScolsep`, 6-1, 6-2, 6-5, 6-9, 6-11, 6-13, 6-15, 6-17
`\DMScounter`, 6-5, 6-9, 6-11, 6-13
`DMSCsequence`, 6-13
`DMSctable`, 6-9
`\DMSform`, 6-5, 6-11
`\DMShead`, 6-1, 6-2, 6-5, 6-9, 6-11, 6-13, 6-15, 6-17, 6-20
`DMSlist`, 6-15
`DMSsequence`, 6-11
`DMStable`, 6-5
`\DMStitle`, 6-19
`\DMStype`, 6-20
`DMSvector`, 6-1
 DOC, 2-1, 2-2, 2-4
`\DOCabstract`, 1-2, 2-2
`\DOCappendix`, 1-2, 2-3

`docbibliography`, 9-1
`\DOCcontents`, 1-2, 2-2
`\DOCfigures`, 1-2, 2-2
`\DOCfloat`, 2-1, 2-2
`\DOCindex`, 1-2, 2-3, 10-2, E-1
`DOCint`, 2-1, 3-5
`docint`, 3-3
`\DOCnumbering`, 1-2, 2-2
`\docpage`, 3-3
`\DOCpreface`, 1-2, 2-2
`\docsec`, 2-2, 3-7
`\DOCTables`, 1-2, 2-2
`\DOCtext`, 1-2, 2-2
 document, 2-1
`\documentstyle`, i, 1-4, E-1
`DOCutilities`, 2-2
`dvihp`, C-1

E

`\E`, 4-7, 7-11
`\E{argd}`, 5-3, 5-7, 6-19
`\E{ARGDES}`, 4-9, 5-3
`\E{argdes}`, 5-3
`\E{argdescription}`, 5-3
`\E{argl}`, 5-2, 5-6, 5-7
`\E{ARGLIST}`, 4-9, 5-2
`\E{arglist}`, 5-2
`\earrow`, 4-1
`\E{Call}`, 5-1, 5-6, 5-7
`\E{call}`, 5-1, 5-7
`\E{deflist}`, 4-9
`\E{DMSarray}`, 6-2
`\E{DMSbook}`, 6-20
`\E{DMSClist}`, 6-17
`\E{DMSCsequence}`, 6-13
`\E{DMSCtable}`, 6-9
`\E{DMSlist}`, 6-15
`\E{DMSsequence}`, 6-11
`\E{DMStable}`, 6-5
`\E{DMSvector}`, 6-1
`\E{errlist}`, 5-6
`\E{listdef}`, 4-9
`\end{argd}`, 5-3, 5-7, 6-19
`\end{ARGDES}`, 4-9, 5-3
`\end{argdes}`, 5-3
`\end{argdescription}`, 5-3
`\end{argl}`, 5-2, 5-6, 5-7
`\end{ARGLIST}`, 4-9, 5-2
`\end{arglist}`, 5-2
`\end{BEcheckOFF}`, 4-7, 7-11
`\end{Block}`, 7-3

`\end{Call}`, 5-1, 5-6, 5-7
`\end{call}`, 5-1, 5-7
`\end{Case}`, 7-6
`\end{deflist}`, 4-9
`\end{DMSarray}`, 6-2
`\end{DMSbook}`, 6-20
`\end{DMSclist}`, 6-17
`\end{DMScsequence}`, 6-13
`\end{DMSctable}`, 6-9
`\end{DMSlist}`, 6-15
`\end{DMSsequence}`, 6-11
`\end{DMStable}`, 6-5
`\end{DMSvector}`, 6-1
`\end{DOC}`, 2-1, 2-2, 2-4
`\end{docbibliography}`, 9-1
`\end{document}`, 2-1
`\end{errlist}`, 5-6
`\end{If}`, 7-6
`\end{listdef}`, 4-9
`\end{Loop}`, 7-4
`\end{Module}`, 7-1
`\end{module}`, 7-1
`\end{Syntax}`, 8-1
`\end{syntax}`, 8-1
Environment
 ARGDES, 4-9, 5-3
 ARGLIST, 4-9, 5-2
 Call, 5-1, 5-6, 5-7
 DMSclist, 6-17
 DMScsequence, 6-13
 DMSctable, 6-9
 DMSarray, 6-2
 DMSbook, 6-20
 DMSlist, 6-15
 DMSsequence, 6-11
 DMStable, 6-5
 DMSvector, 6-1
 argdescription, 5-3
 argdes, 5-3
 argd, 5-3, 5-7, 6-19
 arglist, 5-2
 argl, 5-2, 5-6, 5-7
 call, 5-1, 5-7
 deflist, 4-9
 errlist, 5-6
 listdef, 4-9
 BEcheckOFF, 4-7, 7-11
 Block, 7-3
 Case, 7-6
 DOC, 2-1, 2-2, 2-4
 If, 7-6
 Loop, 7-4
 Module, 7-1

 Syntax, 8-1
 docbibliography, 9-1
 document, 2-1
 module, 7-1
 syntax, 8-1

`\EQ`, 4-1
`\eq`, 4-1
 errlist, 5-6
 Error Description, 5-6

F

`\fboxrule`, 7-2
File
 .dvi, A-1, B-1, C-1
 .hp, C-1, D-1
 .idx, E-1
 .ind, E-1
 .pcl, C-1
 .tex, A-1
 Float parameter, 2-1, 2-2
`\fullpage`, 3-3
 Functions, 5-1

G

`\GE`, 4-1
 german.sty, 1-4
`\GT`, 4-1

H

`\hdots`, 4-1
`\head`, 4-1
`\heute`, 4-8
 HP-Laser, C-1, D-1
 HUGO, 5-7, 6-2 to 6-4, 6-6 to 6-8, 6-10 to 6-12, 6-14, 6-15, 6-19, 7-2, 7-10, 10-2

I

`\I`, 4-8
 IDUM, 6-6
 If, 7-6
`\ifblock`, 7-6
 Index, E-1
`\Index`, 10-2

index
 alphabetical position, 10-2
 entries, 10-1
 subentries, 10-1
`\index`, 10-1
`\Indexbf`, 10-2
`\indexbf`, 10-2
`\indexspace`, 10-2
`\internaldoc`, 3-1
`intes.sty`, i, 10-2
`intes_style`, i
`\inteskopf`, 3-1
`\INTESTITLE`, 3-2
`intex`, A-1
`intexg`, A-1
`\Item`, 10-2
`\item`, 10-2
`\Itembf`, 10-2

K

`\Kasten`, 4-4
`\keyword`, 10-2
`\Kopf`, 3-7

L

laser, D-1
 Laser printer, C-1, D-1
`\lastpage`, 3-7
`latex`, A-1
`\LaTeXfloat`, 2-2
`latexg`, A-1
`\LE`, 4-1
`\leer...\LEER`, 4-8
 Line width, 7-2
 List, 6-15
 list, 4-9, 5-2, 5-3, 5-5, 5-6
`listdef`, 4-9
 Loop, 7-4
`\loopblock`, 7-4
`\LT`, 4-1

M

`makeidx`, E-1
`MakeIndex`, E-1
`makeindex`, E-1
`\makeindex`, E-1
`makeindex.tex`, 10-1, E-1

Module, 7-1
 module, 7-1
 Module block, 7-3
 Module case, 7-6
 Module if-condition, 7-6
 Module loop, 7-4
 Module separator, 7-1, 7-4, 7-6
 Module title, 7-1
 Module width, 7-1, 7-4, 7-6
`\Modulesep`, 7-1, 7-4, 7-6
`\Modulewidth`, 7-1
`myfile.dvi`, A-1, B-1, C-1
`myfile.hp`, C-1, D-1
`myfile.idx`, E-1
`myfile.ind`, E-1
`myfile.pcl`, C-1
`myfile.tex`, A-1

N

`\namlis`, 6-19
`\NE`, 4-1
`\Newlist`, 6-13
`\newlist`, 6-13
`\Newtup`, 6-9
`\newtup`, 6-9
`\note`, 4-4

P

Pagestyle
 AUFTRAG, 3-7
 DOCint, 2-1, 3-5
 docint, 3-3
`\pagestyle`, 2-1, 3-3, 3-5, 3-7
 Path name, 2-4
 preview, B-1
`\printdocindex`, 10-2, E-1
`\printindex`, E-1

R

`\Rahmen`, 4-4
 Revisions, ii

S

`\SboxBCfalse`, 8-1
`\SboxBCtrue`, 8-1

`\Sboxrule`, 8-1
`\Sboxwidth`, 8-1
Screen, B-1
Search path, 2-4
`\Seprow`, 6-2
Sequence, 6-11
`\stackrel`, 4-1
`\subitem`, 10-2
Subroutines, 5-1
`\subsubitem`, 10-2
Symbols, 4-1
Syntax, 8-1
syntax, 8-1
Syntax keyword, 8-1
Syntax structures, 8-1
Syntax title, 8-1
Syntaxbox width, 8-1
`\3`, 4-1

T

Table, 6-5, 8-1
THEINDEX, 10-2
`\titel...\TITEL`, 4-3
`\titlep`, 3-1
`\trademarks`, 3-2

U

`\U`, 4-8
`\uhead`, 4-2

V

Vector, 6-1
`\verbatimfile`, 4-5
`\verbatimlisting`, 4-5

W

`\warrow`, 4-1

X

XRF-Files, 2-4
`\XRFindex`, 2-5
`\XRFinput`, 2-4
`\XRFinputpath`, 2-4
`\XRFlabel`, 2-4
`\XRFpageref`, 2-4
`\XRFref`, 2-4
`\XRFrev`, 2-1, 2-4
`\XRFTitle`, 2-1, 2-4