<pre>print.xtable <- function(x, type = getDetion("xtable.type", "latex").</pre>	lastcol <- rep(" ", nrow(x)+2)	} else { WIDTH <- paste("{", width, "}", sep = "")	<pre>ELABEL <- ">\n" BCAPTION <- paste("<caption "\"="" align='\"",' caption.placement,=""> ",</caption></pre>	varying.digits <- is.matrix(attr(x, "digits", exact = TRUE)) ## Code for lettino "digits" be a matrix was provided by		
<pre>type = getOption("xtable.type", "latex"), file = getOption("xtable.file", ""),</pre>)	sep = "") FCAPTION <- " \n"	## Code for letting "digits" be a matrix was provided by ## Arme Henningsen \(\alpha\) henningsenBagric-econ.uni-kiel.de> ## in e-mail dated 2005-06-01.		
append = getOption("xtable.append", FAISE), floating = getOption("xtable.floating", TRUE),	## booktabs change - Matthieu Stigler: fill the hline arguments ## separately, 1 Feb 2012	BTABULAR <-	ECAPTION <- " \h" BROW <- " <tr>"</tr>	## in e-mail dated 2006-06-04. ##if(!varying. digits) { ## modified Claudio Agostinelli <claudio@unive.it> dated 2006-07-28</claudio@unive.it>		
floating.environment = getOption("xtable.floating.environment", "table"),	## Code before booktabs change was:	paste("\begin{", tabular.environment, "}",	BROW <- " <tr>" EROW <- " </tr> \n" BTH <- " <tb "<="" td=""><td>## modified Claudio Agostinelli <claudio@unive.it> dated 2006-07-28</claudio@unive.it></td></tb>	## modified Claudio Agostinelli <claudio@unive.it> dated 2006-07-28</claudio@unive.it>		
caption.placement = getOption("xtable.caption.placement", "bottom"),	## add.to.rou@pos[[npos+1]] <- hline.after	WIDTH, "{", paste(c(attr(x, "align",	ETH C- 1 C/TH> 1	## $attr(x, "digits") \leftarrow matrix(attr(x, "digits", exact=TRUE),$ ## $nrow = nrow(x)$, $ncol = ncol(x)+1$, $byrow = TRUE)$		
<pre>floating *getuption("ranks.fioking", 1802), floating.extromment *getUption("ranks.fioking.extromment", "table"), table.placement *getUption("ranks.table.placement", "http: caption.placement *getUption("ranks.caption.placement", "bottom"), caption.width * getUption("ranks.caption.width", NULL), itater.extromment *getUption("ranks.caption.width", NULL), itater.extromment *getUption("ranks.caption.width", NULL),</pre>	if (!booktabs){	exact = TRUE)[tmp.index.start:length(attr(x, "align",	STH <- " <th> " BTD1 <- " <td <="" align="" td=""><td><pre>### for(i in 1:ncol(x)) {</pre></td></td></th>	" BTD1 <- " <td <="" align="" td=""><td><pre>### for(i in 1:ncol(x)) {</pre></td></td>	<td><pre>### for(i in 1:ncol(x)) {</pre></td>	<pre>### for(i in 1:ncol(x)) {</pre>
later.extromests "gettpin("raths.later.extromests", c("cate")), tabular extramests gettpin("raths.lather.extromests", "tabular"), tabular extramests gettpin("raths.lather.extromests", "tabular"), tables.after gettpin("raths.later.extromests", "c(-1,0,mred())), talline.after gettpin("raths.later.extromests", TEDD, tabular remains gettpin("raths.later.extromests", TEDD, tabular gettpin("raths.later.extromests.ex	add.to.row\$pos[[npos+1]] <- hline.after	"}\n"), sep = "", collapse = ""), sep = "",	align.tmp<- attr(x, "align", exact = TRUE) align.tmp <- align.tmp[align.tmp!="!"] BTDZ <- matrix(align.tmp[(2-pos):(ncol(x)+1)],	<pre>xcol <- x[, i] if(is.factor(xcol))</pre>		
hline.after = getOption("xtable.hline.after", c(-1,0,nrow(x))),	for(i in i:length(hline.after)) { add.to.row#pos[[mposi]] <- hline.after[i]	sep = "", collapse = ""),	BTD2 <- matrix(align.tmp[(2-pos):(ncol(x)+1)],	xcol <- as.character(xcol) if(is.list(xcol))		
NA.string = getOption("xtable.NA.string", ""), include.rownames = getOption("xtable.include.rownames", TRUE).	add.to.row@pos[[npos+i]] <- hline.after[i]		nrow = nrow(x), ncol = ncol(x)+pos, byrow = INUE)	<pre>if(is.list(xcol)) xcol <- sapply(xcol, unlist)</pre>		
include.colnames = getOption("xtable.include.colnames", TRUE),	} add.to.row\$command <- c(add.to.row\$command, PHEADER)	## fix 10-26-09 (robert.casteloGupf.edu) the following ## 'if' condition is added here to support	## in -mail dated Wednesday, January 17, 2007 BTD2[regemyr("p", BTD2):0] <- "left" BTD2[BTD2 == "r"] <- "right" BTD2[BTD2 == "l"] <- "set"	<pre>if(is.ise(xcol)) xcol <- sapply(xcol, unlist) ina <- is.na(xcol) is.numeric.colum <- is.numeric(xcol)</pre>		
only.contents = getuption("xtable.only.contents", ralse), add.to.row = getOption("xtable.add.to.row", NULL),	add.to.rowscommand <- c(add.to.rowscommand, PHEADER) }	<pre>## 'aj' commattem is added here to support ## a caption on the top of a longtable if (tabular environment == "longtable" && caption.placement == "top") {</pre>	BTD2[BTD2 == "r"] <- "right"			
sanitize.text.function = getOption("xtable.sanitize.text.function", NULL),	if (langth(add to worthcompant) > 0) /	if (tabular.environment = "longtable" && caption.placement == "top") { if (is well(short caption))}	BTD2[BTD2 == "1"] <- "left" PTD2[BTD2 == "c"] <- "conton"	<pre>if(is.character(xcol)) { cols[, i*pos] <- xcol } else {</pre>		
sanitize.text.function),	<pre>if (lamgth(add.to.rowlcommand) > 0) { for (i in !lamgth(add.to.rowlcommand)) { addpos <- add.to.rowlposm([i]) freq (- table(addpos))</pre>	if (is.mull(short.caption)){ BCAPTION <- "\caption("	BTDD (BTD2 == "c"] <- "center" BTD3 <- "\"> " ETD <- " <td>} else { if (is.null(format.args)){</td>	} else { if (is.null(format.args)){		
<pre>sanitize.colnames.function = getOption("xtable.sanitize.colnames.function",</pre>	addpos <- add.to.row8pos[[i]] freq <- table(addpos)	BLAFILON ("\\caption(", short.caption, "]{", sep = "") }		if (is.null(format.args)){ format.args <- list()		
math.style.negative = getOption("xtable.math.style.negative", FALSE),	addpos <- unique(addpos)	}	result <- str	} if (is.null(format.args\$decimal.mark)){		
<pre>sanitize.text.function), nath.style.negative = getOption("ttable.nath.typle.negative", FAISE), html.table.attributes = getOption("ttable.nath.typle.negative", FAISE), html.table.attributes = getOption("ttable.print.results", TRUE), print.results = getOption("ttable.print.results", TRUE),</pre>	rreq - unique(andpo), addpo - unique(addpo) for (j in :length(addpo)) { lastcol[addpos[j]+2] - paste(lastcol[addpos[j]+2],	<pre>FAPTION <- "} \\\\ \n" if ((!is.null(caption)) &k (type == "latex")) {</pre>	result < str ## Changed as suggested in bug report #2795 ## That is replacement of 'R' is 'Ramp;'	<pre>if (is.nuil(format.args@decimal.mark)){ format.args@decimal.mark <- options()@OutDec</pre>		
format.args = getOption("xtable.format.args", NULL), rotate.rownames = getOption("xtable.rotate.rownames", FALSE).	<pre>paste(rep(add.to.row\$command[i],</pre>	BTABULAR <- paste(BTABULAR, BCAPTION, caption, ECAPTION, sep = "")	## instead of previous "kamp" etc ## result <- asub("k", "kamp ", result, fixed = TRUE)) if(!varving.digits)f		
format.args = gstDption("stable.format.args", NULL), rotate.rownmanes = gstDption("stable.rotate.rownmane", FALSE), rotate.colnames = gstDption("stable.rotate.colnames", FALSE), booktable = gstDption("stable.booktable", RALSE),	pasts(rep(add.to.row@command[1],	1	## Inact to replacement of R to amp: ## instead of previous "Mamp" etc ## result <- gash("R", "Mamp", result, fixed = TRUE) ## result <- gash("A", "Mag ", result, fixed = TRUE) ## result <- gash("<", "Rut ", result, fixed = TRUE)	curFormatArgs c(list(
scalebox = getOption("xtable.scalebox", NULL),	sep =)	## Claudio Agostinelli <claudio@unive.it> dated 2006-07-28</claudio@unive.it>	result <- gaub("k", "kamp;", result, fixed = TRUE)			
<pre>scalebox = gstOption("trable.scalebox", WLL), width = gstOption("trable.vidth", WLL), comment = gstOption("trable.comment", TRUE), timestamp = gstOption("trable.cimestamp", date()),</pre>) ·	## add.to.row position -1 BTABULAR <- paste(BTABULAR, lastcol[i], sep = "")	result <- gsub("A", "Ramp;", result, fixed = TRUE) result <- gsub(">", "Agg:", result, fixed = TRUE) result <- gsub(">", "Agg:", result, fixed = TRUE)	format = ifelse(attr(x, "digits", exact = TRUE)[i+1] < 0, "E",		
timestamp = getOption("xtable.timestamp", date()),	if (length(type)>i) stop("\"type\" must have length i")	## the \hline at the end, if present, is set in full matrix ETABULAR <- paste("\end(", tabular.environment, "\n", sep = "")		attr(x, "display", exact = TRUE)[i+1]), digits = abs(attr(x, "digits", exact = TRUE)[i+1])),		
()	<pre>ir (lemgtn(type)>1) stop("\"type\" must nave length 1") type <- tolower(type) if (lall(lis.na(match(type, c("latex","html"))))) {</pre>		## recommended not escaping underscores. ## result < gsub("", "_", result, fixed=TRUE) return(result)	<pre>aigits = abs(attr(x, "aigits", exact = INUE)(1+1j)), format.args) cols(, i*pos] <- do.call("formatC", curFormatArgs)</pre>		
## If caption is length 2, treat the second value as the "short caption" caption <- attr(x, "caption", exact = TRUE)	<pre>if (!all(!is.na(match(type, c("latex","html"))))) { stop("\type\" must be in {\"latex\", \"html\"}")</pre>	## Add scalebox - CR, 7/2/12 if (lis.mull(scalebox)){	1.			
<pre>caption < attr(x, "caption", exact = TRUE) short.caption < NULL if (!is.mull(caption) & length(caption) > 1){ short.caption < caption(2) caption < caption(2)</pre>	## Disabling the check on known floating environments as many users	RTARIII.AR <- maste("\\scalehovi" scalehov "}f\n" RTARIII.AR	sanitize.numbers <- function(x) {	4-4-4-4-4-4-4-1-1-1-1-1-1-1-1-1-1-1-1-1		
if (!is.null(caption) && length(caption) > 1){ short.caption <- caption[2]	## Disabling the check on known floating environments as many users ## want to use additional environments. # if (lall(is.na(match(floating.environment,	sep = "") ETABULAR <- paste(ETABULAR, "}\n", sep = "")	return(x)	for j in throw(cols) / \(\tau\) curFormatrys <- c(list(\(x = \text{xcol}[j]\).		
caption <- caption[1]	# if (!all(!is.na(match(floating.environment, # c("table" "table" "sidemustable"	1	sanitize.final <- function(result) {	x = xcol[j], format =		
1	# c("table", "sable", "satewaystable", # stop("\"type\" must be in {\"able\", \"table\\", \"sidewaystable\\", \"margintable\\"}")	## BSIZE contributed by Benno <pre>puets@mpipsykl.mpg.de> in e-mail</pre>	## Suggested by Use Ligges ligges@statistik.uni-dortmund.de> ## in ==msil dated 2005-07-30. resultBate <_gmai(" ""," ", resultBatext, fixed = TRUE)	ifelse(attr(v "digits" evact = TRUE)[i i+1] < 0		
## Claudio Agostinelli <claudio@unive.it> dated 2006-07-28 hline.after</claudio@unive.it>		## dated Wednesday, December 01, 2004 if (is.mull(size) !is.character(size)) {	result\$text <- gaub(" *", " ", result\$text, fixed = TRUE) result\$text <- gaub(' align="left"', "", result\$text,	"E", attr(x, "display", exact = TRUE)[i+i]),		
## Clausto Agosinesis (commission) to Managara and Africa and Africa (commission) to Managara ## By default if print on Whine before and after the column names ## independently they are printed or not and at the end of the table ## DNL code that act hime. after should include (-1, 0, nrou(a)) in the	if (("margintable" %in% floating.environment)		fixed = TRUE) return(result)	abs(attr(x, "digits", exact = TRUE)[i, i+i])).		
## Old code that set hline.after should include c(-1, 0, nrow(2)) in the ## hline.after vector	in ('margintanis' Ain, loating-environment) & (lia.mull(table.placement)) { warning("margintable does not allow for table placement; setting table.placement to NULL")	ESIZE <- "" } else {	return(result)	format.args) cols[j, i+pos] <- do.call("formatC", curFormatArgs)		
## hine after vector ## If you do not want any \hine inside the data, set hine after to NULL	table.placement <- NULL	if(length(grep("\\\\", size)) == 0){ size <- paste("\\", size, sep = ""))	}		
## PREADER Instead the string '\hitselm' is used in the code ## Now himse after counts how many time a position appear ## I left an automatic PREADER in the longitable is this correct?	if (!is.null(table.placement) &&	1	result <- string("", file = file, append = append)) ·		
	<pre>!all(!is.na(match(unlist(strsplit(table.placement, split = "")),</pre>	BSIZE <- paste("{", size, "\n", sep = "") ESIZE <- "}\n"	info <- R.Version() ## modified Claudio Agostinelli <claudio@unive.it> dated 2006-07-28</claudio@unive.it>	## End Ian Fellows changes		
## Claudio Agostinelli <claudio@unive.it> dated 2006-07-28 include.rounames,</claudio@unive.it>	<pre>c("H',"h","t","b","p","!")))) { stop("\"table.placement\" must contain only elements of {\"h\",\"t\",\"b\",\"p\",\"!\"}")</pre>) PIADET (= "\\label("	## maif/led Clearie Appaired!! <pre>Clearies Appaired!</pre> ## maif/led Clearies Appaired!! <pre>Clearies Appaired!!</pre> ## main to command	if (any(ina)) cols[ina, i+pos] <- MA.string ## Based on contribution from Jonathan Swinton <jonathansswintons.net> ## in e-mail dated Medmeday, January 17, 2007</jonathansswintons.net>		
## include.colnames pos <- 0	<pre>} if (!all(!is.na(match(caption.placement, c("bottom","top"))))) { stop("\"caption.placement\" must be either {\"bottom\",\"top\"}")</pre>		if (comment){ result <- result + BCOMMENT + type + " table generated in " +	## Based on contribution from Jonathan Swinton <jonathandswintons.net> ## in e-mail dated Wednesday, January 17, 2007</jonathandswintons.net>		
pos <- 0 if (include.rownames) pos <- 1	stop("\"caption.placement\" must be either {\"bottom\",\"top\"}")	ELABEL <- "}\n" ## Added caption width (jeff.laake@mooa.gov)	info\$language + " " + info\$major + "." + info\$minor +			
## Claudio Agostinelli <claudio@unive.it> dated 2006-07-28</claudio@unive.it>	*	<pre>if(!is.mull(caption.width)){ BCAPTION <- paste("\parbox{",caption.width,"}{",sep="")</pre>	" package" + ECOMMENT	<pre>cols[, i+pos] <- sanitize.numbers(cols[, i+pos]) } else {</pre>		
<pre>## hline.after checks if (any(hline.after < -1) any(hline.after > nrow(x))) {</pre>	if (type == "latex") { ROTHMENT <- "Y "		<pre>if (!is.null(timestamp)){ result <- result + BCOMMENT + timestamp + ECOMMENT</pre>	<pre>if (is.null(sanitize.text.function)) { cols[, i*pos] <- sanitize(cols[, i*pos])</pre>		
stop("'hline.after' must be inside [-1, nrow(x)]")	BCOMMENT <- "\n" ECOMMENT <- "\n"	} else { BCAFTION <- NULL ECAFTION <- NULL)	<pre>cols(, i*pos) < samitize(cols(, i*pos)) } else { cols(, i*pos) <- samitize.text.function(cols(, i*pos))</pre>		
}	accionami (~ Ton "John S. Wilker " dated 5-19-2003 ## regarding "teafloat" ## See -mail form "Fermando Henrique Fernas P. da Rosa"	ECAPTION <- NOLL	## Claudio Agostinelli <claudio@unive.it> dated 2006-07-28 only.contents</claudio@unive.it>	<pre>cols[, i+pos] <- sanitize.text.function(cols[, i+pos]) }</pre>		
## Claudio Agostinelli <claudio@unive.it> dated 2006-07-28 ## add.to.row checks</claudio@unive.it>	## See e-mail form "Fernando Henrique Ferraz P. da Rosa"	if (is.mull(ahort.caption)){ BCAPTION <- paste(BCAPTION,"\",sep="")	if (!only.comtents) { result <- result + BTABLE) ·		
## add.to.row.checks if (!is.null(add.to.row)) { if (is.list(add.to.row) && length(add.to.row) == 2) {	if (tabular.environment == "longtable" & floating == TRUE) {		result <- result + BENVIRONMENT	,		
<pre>if (is.list(add.to.row) &k length(add.to.row) == 2) { if (is.null(names(add.to.row))) {</pre>	<pre>## cacademicOffgerrar.nei?* dated 10-28-2005 reparting "longitable" if (tabular.eavironemne = "longitable' # floating = TRUED [{</pre>	BCAPTION <- paste(BCAPTION, "\caption[", short.caption, "]{", sep="") }	<pre>if (floating == TRUE) { if ((!is.mull(caption)) &k</pre>	multiplier <- 5 full <- matrix("", nrow = nrow(x), ncol = multiplier*(ncol(x)+pos)+2)		
if (is.mulicames(add.to.row)) { names(add.to.row) - c('pon', 'command') } alse if (any/cort/cames(add.to.row)) = c'(command', 'pon'))) { stop('the names of the elements of 'add.to.row' must be 'pon' and 'command'')		ECAPTION <- paste(ECAPTION,") \n",sep="")	(type == "html" caption.placement == "top")) { result <- result + BCAFTION + caption + ECAFTION	full(, 1) <- BROW full[, multiplier*(0:(ncol(x)+pos-1))+2] <- BROW full[, multiplier*(0:(ncol(x)+pos-1))+2] <- BRD1		
stop("the names of the elements of 'add.to.row' must be 'pos' and 'command'")] if (floating == TRUE) { ## See e-mail from "Pfaff, Bernhard «Bernhard.Pfaff@drku.com>" ## dated 7-09-2007 reparding "suggestion for an amendment of	EROW <- " \\\ \n" BTH <- ""		full[, multiplier*(0:(mcol(x)+pos-1))+3] <- BTD2		
F	## dated 7-09-2003 regarding "suggestion for an amendment of ## the source"	ETH <- ""	if (!is.null(attr(x, "label", exact = TRUE)) && (type == "later" && cantion placement == "top")) {	<pre>full[, multiplier*(0:(ncol(x)+pos-1))+4] <- BTD3 full[, multiplier*(0:(ncol(x)+pos-1))+5] <- cols</pre>		
<pre>if (is.list(add.to.rowSpos) &k in.wector(add.to.rowScommand,</pre>	## the source" ## State of the source ## ## See e-mail from "Mitchell Sudde" ## See e-mail from "Mitchell Suddens.go. unb" dated 2003-07-09 regarding	STH <- " & " BTD1 <- " & "	(type == "latex" && caption.placement == "top")) { result <- result + BLABEL + attr(x, "label", exact = TRUE) + ELABEL	full[, multiplier*(0:(mcol(x)+pos-1))+6] <- ETD		
<pre>if ((npos <- length(add.to.row@pos)) != length(add.to.row@command)) { stop("the length of 'add.to.row@pos' must be equal to the length of 'add.to.</pre>	## «David.Mitchell@dotars.gov.au»" dated 2003-07-09 regarding ## "Additions to R ztable package"	BTD1 <- " & " BTD3 <- ""	attr(x, "label", exact = TRUE) + ELABEL }	full[, multiplier*(mcol(x)+pos)+2] <- paste(EROW, lastcol[-(1:2)],		
stop("the length of 'add.to.row\$pos' must be equal to the length of 'add.to.	accounts.microcrosses.gov.mc "macco accounts of regularing maccounts." The package " .row@command**dec e-mail from 'Garbade, Stews' ### (Command**dec e-mail from 'Garbade, Stews' ### (Command**dec e-mail from 'Garbade, Stews') ### (Command**dec e-mail from 'Garbade, Stews') #### (Command**dec e-mail from 'Garbade, Stews') ##### (Command**dec e-mail from 'Garbade, Stews') ####################################	BTD3 <- ""	} result <- result + BSIZE	sep = " ")		
if (any(unlist(add.to.row\$pos) < -1)	## regarding the floating environment.	ETD <- "" ## Based on contribution from Jonathan Swinton	result <- result + BTABULAR	if (type == "latex") full[, 2] <- ""		
any(unlist(add.to.row\$pos) > nrow(x))) { stop("the values in add.to.row\$pos must be inside the interval [-i, nrow(x)]	BTABLE <- paste("\begin{", floating.environment, "}",)") ifelse(!is_null(table_nlaroment)	## <jonathan@swintons.net> in e-mail dated Wednesday, January 17, 2007 sanitize <- function(str) {</jonathan@swintons.net>	} ## Claudio Acostinelli <claudio@unive.it> dated 2006-07-28</claudio@unive.it>	result <- result + lastcol[2] + paste(t(full), collapse = "") if (lonly contents) {		
}	paste("[", table.placement, "]", sep = ""),	<pre>santing <-function(sr) { result <-first formular formular</pre>	## include.colmames, include.roumames if (include.colmames) { result < result + BROW + BTH	if (lonly.contents) { if (tabular.environment == "longtable") { ## booktabs change added the if() - 1 Feb 2012 if(booktabs) {		
<pre>} else { stop("the first argument ('pos') of 'add.to.row' must be a list, the second argument")</pre>	ument ('comidad')s muski (kata wanteir min mutia) chiracter")	result <- gsub("\\\", "SANITIZE.BACKSLASH", result) result <- gsub("\$", "\\\$", result, fixed = TRUE)	if (include.coinames) (result <- result + BROW + BTH	## booktabs Change added the if() - 1 Feb 2012 if(booktabs) {		
} } else f	(length(later.environments) == 0)) { BENVIRONMENT <- ""	result <- gsub(">", "\$>\$", result, fixed = TRUE) result <- gsub("<" "\$<\$" result fixed = TRUE)	<pre>if (include.rownames) { result <- result + STH</pre>	<pre>if(!booktabs) { result <- result + PHEADER }</pre>		
stop("'add.to.row' argument must be a list of length 2")	EENVIRONMENT <- "" } else {	result <- gsub(" ", "\$ \$", result, fixed = TRUE)	} ## David G. Whiting in e-mail 2007-10-09			
} else {	DENITRONMENT (~ ""	result <- gsub("{", "\\{", result, fixed = TRUE\) result <- gsub("\", "\\\", result, fixed = TRUE\)	## David G. Whiting in e-mail 2007-10-09 if (is.mull(sanitize.colnames.function)) {	## fix 10-27-09 Liviu Andronic (landroninirc@gmail.com) the ## following 'if' condition is inserted in order to avoid ## that bottom caption interferes with a top caption of a longtable		
add.to.row <- list(pos = list(), command = vector(length = 0, mode = "character"))	EEMVIRONMENT <- "" if ("center" %in% latex.environments){	result <- gsub("%", "\\%", result, fixed = TRUE)	## Zerna C. anting in energy 2007-10-09 if (is.mil(amitise.colnames.Tunction)) { CNAMES <- sanitize(names(x)) } else {	## that bottom caption interferes with a top caption of a longtable if (caption.placement == "bottom") {		
npos <- 0	BENVIRONMENT <- paste(BENVIRONMENT, "\centering\n",	result < gumb(", "\"," result, fixed = TRUE) result < gumb("," \"," result, fixed = TRUE) result < gumb("", "\"," result, fixed = TRUE)	CNAMES <- sanitize.colnames.function(names(x))	<pre>if(caption.placement == "bottom"){ if ((!is.null(caption)) && (type == "later")) { result <- result + ECAPTION + caption + ECAPTION</pre>		
}	sep = **)	result <- gsub("#", "\\#", result, fixed = TRUE) result <- gsub(""", "\\verb " ", result, fixed = TRUE)	} if (rotate.colnames) {	result <- result + BCAPTION + caption + ECAPTION		
## Claudio Agostinelli <claudio@unive.it> dated 2006-07-28 add.to.row ## Add further commands at the end of rows</claudio@unive.it>	<pre>} for (i in i:length(latex.environments)) { if (latex.environments[i] == "") next</pre>	result <- gsub("-", "\\-{}", result, fixed = TRUE)	##added by Markus Loscher, 2009-11-16 CNAMES <- paste("\\beginfsideways}", CNAMES, "\\endfsideways}")) if ('is.mull(attr(x, "label", exact = TRUE))) {		
if (type = "latex") {	if (latex.environments[i] =	result <- gssb("SAMITIZE_BACKSLASE", "\$\backslash\$", result, fixed = TRUE) return(result)		<pre>if (!is.mull(attr(x, "label", exact = TRUE))) { result < result + ELABEL + attr(x, "label", exact = TRUE) + ELABEL</pre>		
## Ame Jainshi Combabasi un Euro J 7000 ## (Type == "latex") ## Driginal code before changes in version 1.6-1 ## PREADER <= "likinach"	BENVIRONMENT <- paste(BENVIRONMENT, "", latex.environments[i],		result <- result + paste(CNAMES, collapse = STH)			
## booktabs code from Matthieu Stigler <matthieu.stigler@gmail.com>,</matthieu.stigler@gmail.com>		sanitize.numbers <- function(x) {	result <- result + ETH + ERDW	ETABULAR <- "\\end{longtable}\n"		
## 1 Feb 2012	"}\n", sep = "") EENVIRONMENT <- paste("\ead(", latex.environments[i],	result <- x if (math.style.negative) {	}	result <- result + ETABULAR		
if(booktabs){ PHEADER <- "\\hline\n"	}	## Jake Bowers <jubouers@illinois.edu> in e-mail</jubouers@illinois.edu>	cols <- matrix("", nrow = nrow(x), ncol = ncol(x)+pos)	result <- result + ESIZE		
h atms (F .	## from 2008-08-20 suggested disabling this feature to avoid ## problems with LaTeX's declume package. ## by Florian Wicklensier (florian, wicklensier@un'-tuebingen.de> ## in e-mail from 2008-10-03 requested the ability to use the	if (include.rownames) { ## David G. Whiting in e-mail 2007-10-09	<pre>if (floating == TRUE) { if ((fis.mult(apption)) && (type == "latex" && caption.placement == "botton")) { result <- result + BCAFTION + caption + BCAFTION</pre>		
## This code replaced to fix bug #2309, David Scott, 8 Jan 2014 ## PHEADER <- ifelse(-1%in@hline.after, "\\toprule\n", "")	ETABLE <- paste("\\end{", floating.environment, "}\n", sep = "") } else {	## by Florian Wickelmaier <florian.wickelmaier@uni-tuebingen.de> ## in e-mail from 2008-10-03 requested the ability to use the</florian.wickelmaier@uni-tuebingen.de>	if (is.mull(sanitize.rownames.function)) { RNAMES <- sanitize(row.names(x))	<pre>(type == "latex" && caption.placement == "bottom")) { result <- result + BCAPTION + caption + BCAPTION</pre>		
## if(O%in@hline.after) { ## PHEADER <- c(PHEADER, "\\midrule\n")	FIRST C paste("\end(", floating.environment, ")\n", sep = "") } else { BIRST C "" FIRST C -"	## old behavior.	} else { RNAMES <- samitize.rownames.function(row.names(x)) }	} if (!is.mull(attr(x, "label", exact = TRUE)) &k		
	BENVIRONMENT <- ""	## old behavior. for(i in :lemgth(x)) { result[i] <- gaub("-", "\$-\$", result[i], fixed = TRUE)		<pre>ir (is.mai(atr(x, "label", exact = IRUE)) &E caption.placement == "bottom") { result <- result + ELABEL + atr(x, "label", exact = TRUE) +</pre>		
## jf (nrow(m)%in%hline.after) { ## PHEADER <- c(PHEADER, "\\bottomrule\n")	EENVIRONMENT <- "" }) h	if (rotate.rownames) { ###added by Markus Loscher, 2009-11-16	result <- result + BLABEL + attr(x, "label", exact = TRUE) + ELABEL		
## } if (is.null(hline.after)){		return(result)	RNAMES <- paste("\\begin{sideways}", RNAMES, "\\end{sideways}"))		
DMEADER <- ""	<pre>tmp.index.start <- 1 if (! include.rownames) { while (attr(x, "align", exact = TRUE)[tmp.index.start] == ' ')</pre>	} sanitize.final <- function(result) {	} cols[, 1] <- RNAMES	} result <- result + EENVIRONMENT		
} else {	while (attr(x, "align", exact = TRUE)[tmp.index.start] == ' ')	return(result))	result <- result + ETABLE		
nime.arter <- sort(nime.arter) PREADER <- rep("\nime.arter), length(hline.after)) if (hline.after[i] == -1) {		else {	## Begin vectorizing the formatting code by Ian Fellows [ian@fellstat.com]	result <- samitize.final(result)		
<pre>if (hline.after[1] == -1) { PHEADER[1] <- "\\toprule\n"</pre>	} ## Added "width" argument for use with "tabular*" or	BCOMMENT <- " " ECOMMENT <- " \n"	## 06 Dec 2011	if (print.results){		
} if (hline.after[length(hline.after)] == nrow(x)) {) ## Added "width" argument for use with "tabulars" or ## "tabulars" environments - CR, 7/2/12 If (is.anli(width)) CR CR CR CR CR CR CR C	ETABLE <- "				

\n", sep = "")
ETABLE <- "\n" ## disp <- function(y) { | print(result) || ``` if (hline.after[length(hline.after)] == nrow(x)) { PHEADER[length(hline.after)] <- "\\bottomrule\n" ``` | | ETABLE <- "\m' BENVIRONMENT <- "" EENVIRONMENT <- "" | ## if (is.factor(y)) { ## y <- levels(y)[y] | T . |
}	} else if (is.element(tabular.environment,	EENVIRONMENT <- "" RTARHLAR <- ""	## } ## if (is.list(y)) {	return(invisible(result\$text))
()))	``` } else if (is.elsenn(tabular.environment, ```	ETABULAR <- "" ETABULAR <- ""	## y <- unlist(y)	•
} else { PHEADER <- ""	environments do not support a width specification. Use another environment such as 'tabular*' or 'tabularx' to specify the width.")	BSIZE <- "" ESIZE <- ""	ss } ss return(y)	
}	WIDTH <- ""	BLABEL <- "		