

- s	tatic canDelete()	
- s	tatic canPurge()	
- s	tatic canChild(\$method)	
(;	tatic displayRecursiveItems array \$recursiveItems, belementToDisplay, bool \$display=true)	
lt	tatic getJSCodeToAddFor temChild(\$field_name, \$child_count_js_var)	
F	tatic showAddChildButton ForItemForm(CommonDBTM \$item, \$field_name, \$canedit rnull, \$display=true)	
(;	tatic getItemField \$itemtype) utoinventoryInformation()	
A automiveritorymnormation()		
	NotificationTarget	
	+ \$prefix	
	+ \$table	
	+ \$notification_targets	
	+ \$notification_targets _labels	
	+ \$notificationoptions	
	+ \$data	

	et 6 de plus
+	construct(\$entity =", \$event=", \$object =null, \$options=[])
+	getFromDBForTarget (\$notifications_id, \$type, \$ID)
+	validateSendTo(\$event, array \$infos, \$notify _me=false, \$emitter=null)
+	getSubjectPrefix( \$event=")
+	getContentHeader()
+	getContentFooter()
+	getMessageID()
+	showForNotification (Notification \$notification)
+	add Addition nalInfos For Target ()
+	addAdditionnalUserInfo (array \$data)
	et 35 de plus
+	static getTable(\$classname =null)
+	static getMessageIdForEvent (?string \$itemtype, ?int \$items_id, ?string \$event)

+ \$tag\_descriptions

+ \$dohistory + \$target + \$entity

(string \$itemtype) + static getInstanceByType (\$itemtype, \$event=", \$options=[]) + static updateTargets (\$input)

+ static getTypeName

+ static getInstance (\$item, \$event=", \$options=[])

+ static getInstanceClass

(\$nb=0)

static displayTabContent ForItem(CommonGLPI \$item, \$tabnum=1, \$withtemplate=0) # computeFriendlyName() # isMailMode() # isTargetItemRecursive()

NotificationTargetProject + getEvents() + addAdditionalTargets

(\$event=") + addSpecificTargets (\$data, \$options)

+ getTags()