

## Help

```
#if defined(PremiaCurrentVersion) && PremiaCurrentVersion < (2008+2) //The "#els
#else
/*****
*   CPS - A simple C PDE solver                                     *
*                                                                 *
*   Copyright (c) 2007,                                           *
*   Maya Briani          <m.briani@iac.rm.cnr.it>,               *
*   Francesco Ferreri    <francesco.ferreri@gmail.com>,          *
*   Roberto Natalini     <r.natalini@iac.rm.cnr.it>,              *
*   Marco Papi           <m.papi@iac.rm.cnr.it>                   *
*                                                                 *
*****/
#ifndef GRID_NODE_H
#define GRID_NODE_H

#include "
href../../../../common/math/highdim_solver/cps_types_h_src.pdfcps_types.h"
#include "
href../../../../common/math/highdim_solver/cps_dimensions_h_src.pdfcps_dimensions.h"
#include "
href../../../../common/math/highdim_solver/cps_grid_h_src.pdfcps_grid.h"

struct grid_node_t
{

    const grid    *source_grid;
    int  tick[MAX_DIMENSIONS];
    double  value[MAX_DIMENSIONS];
    unsigned int order;
};

int grid_node_create(grid_node **);
int grid_node_destroy(grid_node **);
int grid_node_is_left_boundary(const grid_node *, int dim);
int grid_node_is_right_boundary(const grid_node *, int dim);
int grid_node_is_boundary(const grid_node *);
int grid_node_is_external(const grid_node *);
int grid_node_is_internal(const grid_node *);
int grid_node_is_initial(const grid_node *);
```

```
int grid_node_is_final(const grid_node *);  
int grid_node_is_guard(const grid_node *);  
int grid_node_time_forth(grid_node *);  
int grid_node_time_back(grid_node *);  
#endif  
  
#endif //PremiaCurrentVersion
```