

Parallel pagerank

0.1

Generated by Doxygen 1.8.15



<b>1 Data Structure Index</b>	<b>1</b>
1.1 Data Structures	1
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 Data Structure Documentation</b>	<b>5</b>
3.1 NTWPR_CRS Struct Reference	5
3.1.1 Detailed Description	5
3.1.2 Field Documentation	5
3.1.2.1 col_ind	5
3.1.2.2 edge_num	6
3.1.2.3 node_num	6
3.1.2.4 row_ptr	6
3.1.2.5 val	6
3.2 NTWPR_WGFile Struct Reference	6
3.2.1 Detailed Description	6
3.2.2 Field Documentation	7
3.2.2.1 edge_data	7
3.2.2.2 edge_num	7
3.2.2.3 node_num	7
<b>4 File Documentation</b>	<b>9</b>
4.1 /home/ecat/nikos/parallel/source/helpers/ntwpr_wg.c File Reference	9
4.1.1 Detailed Description	9
4.1.2 Function Documentation	10
4.1.2.1 NTWPR_expfm()	10
4.1.2.2 NTWPR_load2crs()	10
4.1.2.3 NTWPR_SU2WG()	10
4.1.2.4 NTWPR_WGfclose()	11
4.1.2.5 NTWPR_WGFile_Reset()	11
4.1.2.6 NTWPR_WGfopen()	11
4.2 /home/ecat/nikos/parallel/source/helpers/ntwpr_wg.h File Reference	12
4.2.1 Detailed Description	13
4.2.2 Macro Definition Documentation	13
4.2.2.1 DF_SU_LINE_COMMENT	13
4.2.2.2 NTWPR_invalid_fp_exit	14
4.2.3 Typedef Documentation	15
4.2.3.1 NTWPR_CRS	15
4.2.3.2 NTWPR_WGFile	15
4.2.4 Function Documentation	15
4.2.4.1 NTWPR_expfm()	15
4.2.4.2 NTWPR_load2crs()	16

4.2.4.3 NTWPR_SU2WG()	16
4.2.4.4 NTWPR_WGfclose()	16
4.2.4.5 NTWPR_WGFile_Reset()	17
4.2.4.6 NTWPR_WGfopen()	17
4.3 /home/ecat/nikos/parallel/source/helpers/test.c File Reference	17
4.3.1 Detailed Description	18
<b>Index</b>	<b>19</b>

# Chapter 1

## Data Structure Index

### 1.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">NTWPR_CRS</a>	
A Compressed Row Storage struct . . . . .	5
<a href="#">NTWPR_WGFile</a>	
A struct containing a web-graph file and the graphs info . . . . .	6



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all documented files with brief descriptions:

/home/ecat/nikos/parallel/source/helpers/ <a href="#">ntwpr_wg.c</a>	
File containing the implementations of <a href="#">ntwpr_wg.h</a>	9
/home/ecat/nikos/parallel/source/helpers/ <a href="#">ntwpr_wg.h</a>	
Web-Graph import/export functions	12
/home/ecat/nikos/parallel/source/helpers/ <a href="#">test.c</a>	
Test script used for development	17





## Chapter 3

# Data Structure Documentation

### 3.1 NTWPR\_CRS Struct Reference

A Compressed Row Storage struct.

```
#include <ntwpr_wg.h>
```

#### Data Fields

- uint32\_t [edge\\_num](#)
- uint32\_t [node\\_num](#)
- double \* [val](#)
- uint32\_t \* [col\\_ind](#)
- uint32\_t \* [row\\_ptr](#)

#### 3.1.1 Detailed Description

A Compressed Row Storage struct.

Used to save web graph tables in RAM. See [CRS](#)

#### 3.1.2 Field Documentation

##### 3.1.2.1 col\_ind

```
uint32_t* NTWPR_CRS::col_ind
```

A vector containing the columns of the val's values.

### 3.1.2.2 edge\_num

```
uint32_t NTWPR_CRS::edge_num
```

The number of non zero elements (edges)

### 3.1.2.3 node\_num

```
uint32_t NTWPR_CRS::node_num
```

The number of nodes of the graph.

### 3.1.2.4 row\_ptr

```
uint32_t* NTWPR_CRS::row_ptr
```

A pointer to the val vector indicating the start of a matrix' row.

### 3.1.2.5 val

```
double* NTWPR_CRS::val
```

The non zero values of the matrix.

The documentation for this struct was generated from the following file:

- [/home/ecat/nikos/parallel/source/helpers/ntwpr\\_wg.h](/home/ecat/nikos/parallel/source/helpers/ntwpr_wg.h)

## 3.2 NTWPR\_WGFile Struct Reference

A struct containing a web-graph file and the graphs info.

```
#include <ntwpr_wg.h>
```

### Data Fields

- `uint32_t` [edge\\_num](#)
- `uint32_t` [node\\_num](#)
- `FILE *` [edge\\_data](#)

### 3.2.1 Detailed Description

A struct containing a web-graph file and the graphs info.

Used as a "middle" save format. This API uses this type to compute the pagerank of the nodes. Any dataset's pagerank vector can be computed by extracting such struct from the dataset.

## 3.2.2 Field Documentation

### 3.2.2.1 edge\_data

FILE\* NTWPR\_WGFile::edge\_data

The edges represented as node pairs

### 3.2.2.2 edge\_num

uint32\_t NTWPR\_WGFile::edge\_num

The number of the graph's edges.

### 3.2.2.3 node\_num

uint32\_t NTWPR\_WGFile::node\_num

The number of the graph's nodes.

The documentation for this struct was generated from the following file:

- [/home/ecat/nikos/parallel/source/helpers/ntwpr\\_wg.h](#)



# Chapter 4

## File Documentation

### 4.1 /home/ecat/nikos/parallel/source/helpers/ntwpr\_wg.c File Reference

File containing the implementations of [ntwpr\\_wg.h](#).

```
#include "ntwpr_wg.h"
```

#### Functions

- [NTWPR\\_CRS](#) \* [NTWPR\\_load2crs](#) ([NTWPR\\_WGFile](#) \*restrict [NTWPR\\_in\\_fp](#))  
*Loads a web graph in memory using the CRS data structure.*
- [NTWPR\\_WGFile](#) \* [NTWPR\\_WGfopen](#) (const char path[static 1])  
*Creates a new WGFile at path.*
- int [NTWPR\\_WGfclose](#) ([NTWPR\\_WGFile](#) \*wgfile)  
*Closes the [NTWPR\\_WGFile](#) stream and frees memory.*
- bool [NTWPR\\_WGFile\\_Reset](#) ([NTWPR\\_WGFile](#) \*const [NTWPR\\_WGF](#))  
*Reset the [NTWPR\\_WGF](#) struct to read the edges again.*
- void [NTWPR\\_expfm](#) ([NTWPR\\_WGFile](#) \*restrict wgfp, const char exp\_path[static 1], uint32\_t [NTWPR\\_↔](#) node\_num)  
*Exports the full graph matrix of a [NTWPR\\_WGFile](#) to the file at exp\_path.*
- void [NTWPR\\_SU2WG](#) (const char SU\_WGD\_path[static 1], const char exp\_path[static 1], uint32\_t [NTWP↔](#) R\_node\_num)  
*Converts web graph files from the format Stanford U. (SU) used to the [NTWPR\\_WGFile](#)' edge\_data file format.*

#### 4.1.1 Detailed Description

File containing the implementations of [ntwpr\\_wg.h](#).

#### Author

Katomeris Nikolaos, 8551, ngkatomer@auth.gr

#### Date

09-08-2018

## 4.1.2 Function Documentation

### 4.1.2.1 NTWPR\_expfm()

```
void NTWPR_expfm (
    NTWPR_WGFile *restrict NTWPR_WGF,
    const char exp_path[static 1],
    uint32_t NTWPR_node_num )
```

Exports the full graph matrix of a [NTWPR\\_WGFile](#) to the file at *exp\_path*.

Should be used with caution and for relatively small *NTWPR\_node\_num*. ( $NTWPR\_node\_num < 10000$ )

#### Parameters

<i>NTWPR_WGF</i>	The input <a href="#">NTWPR_WGFile</a> pointer.
<i>exp_path</i>	The output file's path.
<i>NTWPR_node_num</i>	The number of nodes that will be in the graph matrix.

### 4.1.2.2 NTWPR\_load2crs()

```
NTWPR_CRS* NTWPR_load2crs (
    NTWPR_WGFile *restrict NTWPR_in_fp )
```

Loads a web graph in memory using the CRS data structure.

#### Parameters

<i>NTWPR_in_↔ _fp</i>	The input web graph data.
---------------------------	---------------------------

#### Returns

NTWPR\_CRS\* pointing to the loaded CRS or null.

### 4.1.2.3 NTWPR\_SU2WG()

```
void NTWPR_SU2WG (
    const char SU_WGD_path[static 1],
    const char exp_path[static 1],
    uint32_t NTWPR_node_num )
```

Converts web graph files from the format Stanford U. (SU) used to the [NTWPR\\_WGFile](#) edge\_data file format.

Extra: The output can be loaded into MATLAB with: `fread(file_id, [2,inf], 'uint32');`

## Parameters

<i>SU_WGD_path</i>	The input file's path..
<i>exp_path</i>	The output file's path.
<i>NTWPR_node_num</i>	The number of nodes that will be in the graph matrix. (Enter 0 to get all the nodes of the file.)

## 4.1.2.4 NTWPR\_WGfclose()

```
int NTWPR_WGfclose (
    NTWPR_WGFile * wgfile )
```

Closes the [NTWPR\\_WGFile](#) stream and frees memory.

## Parameters

<i>wgfile</i>	The file to be closed.
---------------	------------------------

## Returns

int Returns 0 if succussful. On failure EOF is returned.

## 4.1.2.5 NTWPR\_WGFile\_Reset()

```
bool NTWPR_WGFile_Reset (
    NTWPR_WGFile *const NTWPR_WGF )
```

Reset the *NTWPR\_WGF* struct to read the edges again.

## Parameters

<i>NTWPR_WGF</i>	The <a href="#">NTWPR_WGFile</a> struct.
------------------	--

## Returns

true If successful.  
false If an error occured.

## 4.1.2.6 NTWPR\_WGfopen()

```
NTWPR_WGFile* NTWPR_WGfopen (
    const char path[static 1] )
```

Creates a new WGFile at *path*.

The WGFile should be then closed with [NTWPR\\_WGfclose\(\)](#)

#### Parameters

<i>path</i>	The path of the file.
-------------	-----------------------

#### Returns

NTWPR\_WGFile\* The struct containing the file created.

## 4.2 /home/ecat/nikos/parallel/source/helpers/ntwpr\_wg.h File Reference

Web-Graph import/export functions.

```
#include <stdio.h>
#include <stdbool.h>
#include <stdlib.h>
#include <stdint.h>
```

### Data Structures

- struct [NTWPR\\_CRS](#)  
*A Compressed Row Storage struct.*
- struct [NTWPR\\_WGFile](#)  
*A struct containing a web-graph file and the graphs info.*

### Macros

- #define [NTWPR\\_WG\\_H](#) 1  
*Macro used to not redefine header definitions when this file is included to other sources.*
- #define [DF\\_SU\\_LINE\\_COMMENT](#) '#'  
*The character used to indicate comment lines in the Stanford's (SU) data files.*
- #define [NTWPR\\_invalid\\_fp\\_exit](#)(fp)  
*Checks for null file pointers returning debug message.*

### Typedefs

- typedef struct [NTWPR\\_CRS](#) [NTWPR\\_CRS](#)  
*A Compressed Row Storage struct.*
- typedef struct [NTWPR\\_WGFile](#) [NTWPR\\_WGFile](#)  
*A struct containing a web-graph file and the graphs info.*



## Functions

- [NTWPR\\_WGFile](#) \* [NTWPR\\_WGfopen](#) (const char path[static 1])  
*Creates a new WGFile at path.*
- int [NTWPR\\_WGfclose](#) ([NTWPR\\_WGFile](#) \*wgfile)  
*Closes the [NTWPR\\_WGFile](#) stream and frees memory.*
- bool [NTWPR\\_WGFile\\_Reset](#) ([NTWPR\\_WGFile](#) \*const [NTWPR\\_WGF](#))  
*Reset the [NTWPR\\_WGF](#) struct to read the edges again.*
- [NTWPR\\_CRS](#) \* [NTWPR\\_load2crs](#) ([NTWPR\\_WGFile](#) \*restrict [NTWPR\\_in\\_fp](#))  
*Loads a web graph in memory using the CRS data structure.*
- void [NTWPR\\_SU2WG](#) (const char SU\_WGD\_path[static 1], const char exp\_path[static 1], uint32\_t [NTWPR\\_node\\_num](#))  
*Converts web graph files from the format Stanford U. (SU) used to the [NTWPR\\_WGFile](#)' edge\_data file format.*
- void [NTWPR\\_expfm](#) ([NTWPR\\_WGFile](#) \*restrict [NTWPR\\_WGF](#), const char exp\_path[static 1], uint32\_t [NTWPR\\_node\\_num](#))  
*Exports the full graph matrix of a [NTWPR\\_WGFile](#) to the file at exp\_path.*

### 4.2.1 Detailed Description

Web-Graph import/export functions.

Currently supported data:

- Files containing sorted edges of web graphs like the ones found at: [Stanford Large Network Dataset Collection](#)

#### Author

Katomeris Nikolaos

#### Date

09-08-2018

### 4.2.2 Macro Definition Documentation

#### 4.2.2.1 DF\_SU\_LINE\_COMMENT

```
#define DF_SU_LINE_COMMENT '#'
```

The character used to indicate comment lines in the Stanford's (SU) data files.

#### 4.2.2.2 NTWPR\_invalid\_fp\_exit

```
#define NTWPR_invalid_fp_exit(  
    fp )
```

**Value:**

```
do {  
    if (!fp)  
    {  
        fprintf(stderr,  
            "%s:%d: Invalid file pointer.\n", __FILE__, __LINE__);  
        exit(EXIT_FAILURE);  
    }  
} while(false)
```

Checks for null file pointers returning debug message.

**Parameters**

<i>fp</i>	The file pointer to check for null.
-----------	-------------------------------------

**4.2.3 Typedef Documentation****4.2.3.1 NTWPR\_CRS**

```
typedef struct NTWPR_CRS NTWPR_CRS
```

A Compressed Row Storage struct.

Used to save web graph tables in RAM. See CRS

**4.2.3.2 NTWPR\_WGFile**

```
typedef struct NTWPR_WGFile NTWPR_WGFile
```

A struct containing a web-graph file and the graphs info.

Used as a "middle" save format. This API uses this type to compute the pagerank of the nodes. Any dataset's pagerank vector can be computed by extracting such struct from the dataset.

**4.2.4 Function Documentation****4.2.4.1 NTWPR\_expfm()**

```
void NTWPR_expfm (
    NTWPR_WGFile *restrict NTWPR_WGF,
    const char exp_path[static 1],
    uint32_t NTWPR_node_num )
```

Exports the full graph matrix of a [NTWPR\\_WGFile](#) to the file at *exp\_path*.

Should be used with caution and for relatively small *NTWPR\_node\_num*. (*NTWPR\_node\_num* < 10000)

**Parameters**

<i>NTWPR_WGF</i>	The input <a href="#">NTWPR_WGFile</a> pointer.
<i>exp_path</i>	The output file's path.
<i>NTWPR_node_num</i>	The number of nodes that will be in the graph matrix.

#### 4.2.4.2 NTWPR\_load2crs()

```
NTWPR_CRS* NTWPR_load2crs (
    NTWPR_WGFile *restrict NTWPR_in_fp )
```

Loads a web graph in memory using the CRS data structure.

##### Parameters

<i>NTWPR_in_fp</i>	The input web graph data.
--------------------	---------------------------

##### Returns

NTWPR\_CRS\* pointing to the loaded CRS or null.

#### 4.2.4.3 NTWPR\_SU2WG()

```
void NTWPR_SU2WG (
    const char SU_WGD_path[static 1],
    const char exp_path[static 1],
    uint32_t NTWPR_node_num )
```

Converts web graph files from the format Stanford U. (SU) used to the [NTWPR\\_WGFile](#)' edge\_data file format.

Extra: The output can be loaded into MATLAB with: `fread(file_id, [2,inf], 'uint32')`;

##### Parameters

<i>SU_WGD_path</i>	The input file's path..
<i>exp_path</i>	The output file's path.
<i>NTWPR_node_num</i>	The number of nodes that will be in the graph matrix. (Enter 0 to get all the nodes of the file.)

#### 4.2.4.4 NTWPR\_WGfclose()

```
int NTWPR_WGfclose (
    NTWPR_WGFile * wgfile )
```

Closes the [NTWPR\\_WGFile](#) stream and frees memory.

## Parameters

<code>wgfile</code>	The file to be closed.
---------------------	------------------------

## Returns

int Returns 0 if succussful. On failure EOF is returned.

## 4.2.4.5 NTWPR\_WGFile\_Reset()

```
bool NTWPR_WGFile_Reset (
    NTWPR_WGFile *const NTWPR_WGF )
```

Reset the *NTWPR\_WGF* struct to read the edges again.

## Parameters

<code>NTWPR_WGF</code>	The <code>NTWPR_WGFile</code> struct.
------------------------	---------------------------------------

## Returns

true If successful.  
false If an error occured.

## 4.2.4.6 NTWPR\_WGfopen()

```
NTWPR_WGFile* NTWPR_WGfopen (
    const char path[static 1] )
```

Creates a new WGFile at *path*.

The WGFile should be then closed with `NTWPR_WGfclose()`

## Parameters

<code>path</code>	The path of the file.
-------------------	-----------------------

## Returns

NTWPR\_WGFile\* The struct containing the file created.

## 4.3 /home/ecat/nikos/parallel/source/helpers/test.c File Reference

Test script used for development.

```
#include <stdio.h>
#include <stdlib.h>
#include "ntwpr_wg.h"
```

## Functions

- int **main** (int argc, char \*argv[argc+1])

### 4.3.1 Detailed Description

Test script used for development.

#### Author

Katomeris Nikolaos, 8551, ngkatomer@auth.gr

#### Date

09-08-2018

# Index

/home/ecat/nikos/parallel/source/helpers/ntwpr\_wg.c, [9](#)  
/home/ecat/nikos/parallel/source/helpers/ntwpr\_wg.h,  
[12](#)  
/home/ecat/nikos/parallel/source/helpers/test.c, [17](#)

col\_ind  
    NTWPR\_CRS, [5](#)

DF\_SU\_LINE\_COMMENT  
    ntwpr\_wg.h, [13](#)

edge\_data  
    NTWPR\_WGFile, [7](#)

edge\_num  
    NTWPR\_CRS, [5](#)  
    NTWPR\_WGFile, [7](#)

NTWPR\_CRS, [5](#)  
    col\_ind, [5](#)  
    edge\_num, [5](#)  
    node\_num, [6](#)  
    ntwpr\_wg.h, [15](#)  
    row\_ptr, [6](#)  
    val, [6](#)

NTWPR\_SU2WG  
    ntwpr\_wg.c, [10](#)  
    ntwpr\_wg.h, [16](#)

NTWPR\_WGFile, [6](#)  
    edge\_data, [7](#)  
    edge\_num, [7](#)  
    node\_num, [7](#)  
    ntwpr\_wg.h, [15](#)

NTWPR\_WGFile\_Reset  
    ntwpr\_wg.c, [11](#)  
    ntwpr\_wg.h, [17](#)

NTWPR\_WGfclose  
    ntwpr\_wg.c, [11](#)  
    ntwpr\_wg.h, [16](#)

NTWPR\_WGfopen  
    ntwpr\_wg.c, [11](#)  
    ntwpr\_wg.h, [17](#)

NTWPR\_expfm  
    ntwpr\_wg.c, [10](#)  
    ntwpr\_wg.h, [15](#)

NTWPR\_invalid\_fp\_exit  
    ntwpr\_wg.h, [13](#)

NTWPR\_load2crs  
    ntwpr\_wg.c, [10](#)  
    ntwpr\_wg.h, [16](#)

node\_num

NTWPR\_CRS, [6](#)  
NTWPR\_WGFile, [7](#)

ntwpr\_wg.c  
    NTWPR\_SU2WG, [10](#)  
    NTWPR\_WGFile\_Reset, [11](#)  
    NTWPR\_WGfclose, [11](#)  
    NTWPR\_WGfopen, [11](#)  
    NTWPR\_expfm, [10](#)  
    NTWPR\_load2crs, [10](#)

ntwpr\_wg.h  
    DF\_SU\_LINE\_COMMENT, [13](#)  
    NTWPR\_CRS, [15](#)  
    NTWPR\_SU2WG, [16](#)  
    NTWPR\_WGFile, [15](#)  
    NTWPR\_WGFile\_Reset, [17](#)  
    NTWPR\_WGfclose, [16](#)  
    NTWPR\_WGfopen, [17](#)  
    NTWPR\_expfm, [15](#)  
    NTWPR\_invalid\_fp\_exit, [13](#)  
    NTWPR\_load2crs, [16](#)

row\_ptr  
    NTWPR\_CRS, [6](#)

val  
    NTWPR\_CRS, [6](#)