

---

# Nintendo Wi-Fi Connection

## TWL-DWC Programming Manual: General-Purpose Ranking Edition

Version 2.1.1

**The content of this document is highly confidential  
and should be handled accordingly.**

**Confidential**

These coded instructions, statements, and computer programs contain proprietary information of Nintendo and/or its licensed developers and are protected by national and international copyright laws. They may not be disclosed to third parties or copied or duplicated in any form, in whole or in part, without the prior written consent of Nintendo.

## Table of Contents

1	Introduction .....	7
2	How to Use the Library .....	8
2.1	Process Flow .....	8
2.1.1	Initializing the General-Purpose Ranking Library .....	9
2.1.2	Uploading Scores .....	9
2.1.3	Downloading Scores .....	10
2.1.4	Polling the Asynchronous Process to Get the Communication Result Status .....	16
2.2	Terminating the General-Purpose Ranking Library .....	17
2.3	Canceling a Process and Error Handling .....	17
2.4	About the rank_sample Sample Program .....	17
2.5	Amount of Memory Used .....	18
3	General-Purpose Ranking Management Tool .....	19
3.1	Overview .....	19
3.2	File Structure .....	19
3.3	Execution Environment .....	19
3.3.1	Install .NET Framework .....	19
3.3.2	Install the Settings File .....	20
3.3.3	Configure Communications .....	20
3.4	Communications Load on the General-Purpose Ranking Server .....	20
3.5	Time Zones of Acquired Data .....	20
3.6	Starting the General-Purpose Ranking Management Tool .....	20
3.7	Screen Structure .....	21
3.8	Menus .....	22
3.8.1	File Menu .....	22
3.8.2	Edit Menu .....	22
3.8.3	Operation Menu .....	23
3.8.4	Settings Menu .....	23
3.8.5	Help Menu .....	23
3.9	Dialog Boxes .....	23
3.9.1	Get TSV Dialog Box .....	23
3.9.2	Get UserData Dialog Box .....	24
3.9.3	Delete Entry Dialog Box .....	25
3.9.4	Proxy Server Settings Dialog Box .....	26
3.10	Shortcut Keys .....	26
4	Web Service Development .....	27
4.1	About Web Services .....	27

4.2	How to Use .....	27
4.2.1	How to Access .....	27
4.2.2	About Security.....	27
4.2.3	About the Data Format.....	27
4.2.4	About Server Loads .....	27
4.2.5	Get Parameters.....	28

## Code

Code 2-1	Initialization of the Ranking Library .....	9
Code 2-2	Request to Start Uploading Scores .....	10
Code 2-3	Request to Start Score Download to Get Rank Order.....	13
Code 2-4	Request to Start Score Download to Get Top-Ranking List .....	14
Code 2-5	Accessing the Obtained Ranking List .....	15
Code 2-6	Asynchronous Process Polling .....	16

## Tables

Table 2-1	Get Modes for Downloading Scores.....	11
Table 2-2	Size Specified in Parameter for Each Get Mode.....	11
Table 2-3	Parameters Specified for Rank Order Get Mode .....	12
Table 2-4	Parameters Specified for Top-Rankings List Get Mode .....	12
Table 2-5	Parameters Specified for the Near-Rankings List Get Mode .....	12
Table 2-6	Parameters Specified for the Friend-Rankings List Get Mode .....	13

## Figures

Figure 2-1	General-Purpose Ranking Library Process Flow.....	8
Figure 3-1	Organization.....	19
Figure 3-2	General-Purpose Ranking Management Tool Main Window .....	21
Figure 3-3	Get TSV Dialog Box .....	23
Figure 3-4	Get UserData Dialog Box.....	24
Figure 3-5	Delete Entry Dialog Box.....	25
Figure 3-6	Proxy Server Settings Dialog Box.....	26

## Revision History

Version	Revision Date	Description
2.1.1	2009/08/20	Revised section 2.1.1 Initializing the General-Purpose Ranking Library to state that switching the authentication server can now be accomplished using the DWC initialization functions <code>DWC_InitForDevelopment</code> or <code>DWC_InitForProduction</code> .
2.1.0	2009/04/24	In sections 3.5 Time Zones of Acquired Data, 3.8.3 Operation Menu, 3.9.1 Get TSV Dialog Box, and 3.10 Shortcut Keys, the files that can be obtained from the general-purpose ranking management tool are tab-delimited (TSV) rather than comma-delimited (CSV), so the tool's interface has been revised. Accordingly, all related items in this document have been revised in the same way.  Revised explanations in sections 4.2.1 How to Access and 4.2.3 About the Data Format because the files that can be obtained are tab-delimited (TSV) rather than comma-delimited (CSV).  Supported the added region definition for China in sections 2.1.2 Uploading Scores, 3.7 Screen Structure, and 4.2.5 Get Parameters.
2.0.2	2008/12/04	In section 2.1.3.1 How to Specify the DWCRnkGetParam Structure: Corrected to state that it is possible to get a maximum of 65 pieces of data in the Near-Rankings List Get mode.
2.0.1	2008/11/28	Deleted related description in line with making the GHTTP library private. (* Unreleased)
2.0.0	2008/10/10	Transitioned from NITRO-DWC to TWL-DWC.
1.4.0	2008/09/06	Released with NITRO-DWC 3.1 plus2.
1.3.3	2008/09/05	Deleted mention of submission of design statement from section 2.1.1. (Unreleased)
1.3.2	2008/08/29	In section 2.1.1, changed the address of the DNS server for changing from the production server to the development server.  New address: 125.206.241.210 Old address: 125.206.241.190 (Unreleased)
1.3.1	2008/08/07	In section 4.2.5, added mention that zero padding is not permitted when the <code>since</code> parameter is specified. (Unreleased)
1.3.0	2008/06/10	Released with the official version of NITRO-DWC 3.1.
1.2.2	2008/04/17	Added a note to section 3.9.3 Delete Entry Dialog Box explaining that with the production server you cannot delete by specifying <b>all</b> in the <b>Delete Entry</b> dialog box. ( <b>Note:</b> Version not released as a separate update.)
1.2.1	2008/03/27	Updated the titles of referenced documents. (Note: Version not released as a separate update.)
1.2.0	2008/03/03	Released with NITRO-DWC 3.0 plus patch.
1.1.3	2008/01/23	Added description to section 2.1.1 regarding how to connect from commercial product to the general-purpose ranking development server.

Version	Revision Date	Description
		(Note: Version not released as a separate update.)
1.1.2	2008/01/15	Added an explanation to section 2.1.3.3 regarding the <b>order</b> field in the get mode for top ranking lists. ( <b>Note:</b> Version not released as a separate update.)
1.1.1	2008/01/07	Revised in line with the implementation of timeout in the library. Released with official version of NITRO-DWC 3.0.
1.1.0	2007/07/19	Revised the maximum number of categories from 100 to 1000 in section 2.1.2. Revised the maximum number of rankings that can be obtained at one time from 10 to 30 in section 2.1.3.1. Added a description to section 2.1.3.1 regarding high and low near-rankings acquisition. Added a description to section 3.6 regarding the management tool's time zone settings. Added a description to sections 3.8.4 and 3.9.4 regarding the management tool's proxy settings.
1.0.9	2007/04/27	Added support for the Korean market.
1.0.8	2007/04/18	Revised the size of user-defined data to 764 bytes in section 2.1.2.
1.0.7	2007/04/10	Added a note to Table 2-3 regarding the effect of getting one's own ranking when it is specified. Added a note to section 2.1.3.3 regarding the effect of getting one's own score when it is specified.
1.0.6	2007/02/02	Corrected a typo in section 3.8.1 regarding base64 encoding. Corrected a typo in section 4.2.3 regarding base64 encoding.
1.0.5	2006/09/20	Added a note to section 3.8.1 that base64 encoding is a proprietary format. Added a note to section 3.8.3 that in the development server, all PIDs may be deleted at once. Revised the URL where the CSV file is retrieved in section 4.2.1. Added a note to section 4.2.3 that base64 encoding is a proprietary format.
1.0.4	2006/07/26	Revised text specific to section 4.2.3 CSV File Formats.
1.0.3	2006/07/21	In section 2.1.3.3, added a note and revised notes specific to cases where ranking order cannot be identified.
1.0.2	2006/06/19	In section 2.1.3.3, added notes specific to cases where ranking order cannot be identified.
1.0.1	2006/06/12	Corrected errors in section 3.6 Screen Composition. Corrected errors in section 3.8.3 Delete Entry Dialog Box.
1.0.0	2006/06/06	Initial version.

# 1 Introduction

This document explains how to use the general-purpose ranking library of the TWL-DWC and its related administration tool. It also explains the Web development interface.

The following features are provided by the general-purpose ranking library.

- Upload scores
- Get one's own ranking order
- Get the top ranking list
- Get a ranking list of players with scores near one's own score
- Get lists of ranking among friends and among rivals

To implement these features, the general-purpose ranking library provides a set of APIs for communicating with the general-purpose ranking server.

To read more about the actual use of this library, see the *Nintendo Wi-Fi Connection Programming Guidelines for Nintendo DS*.

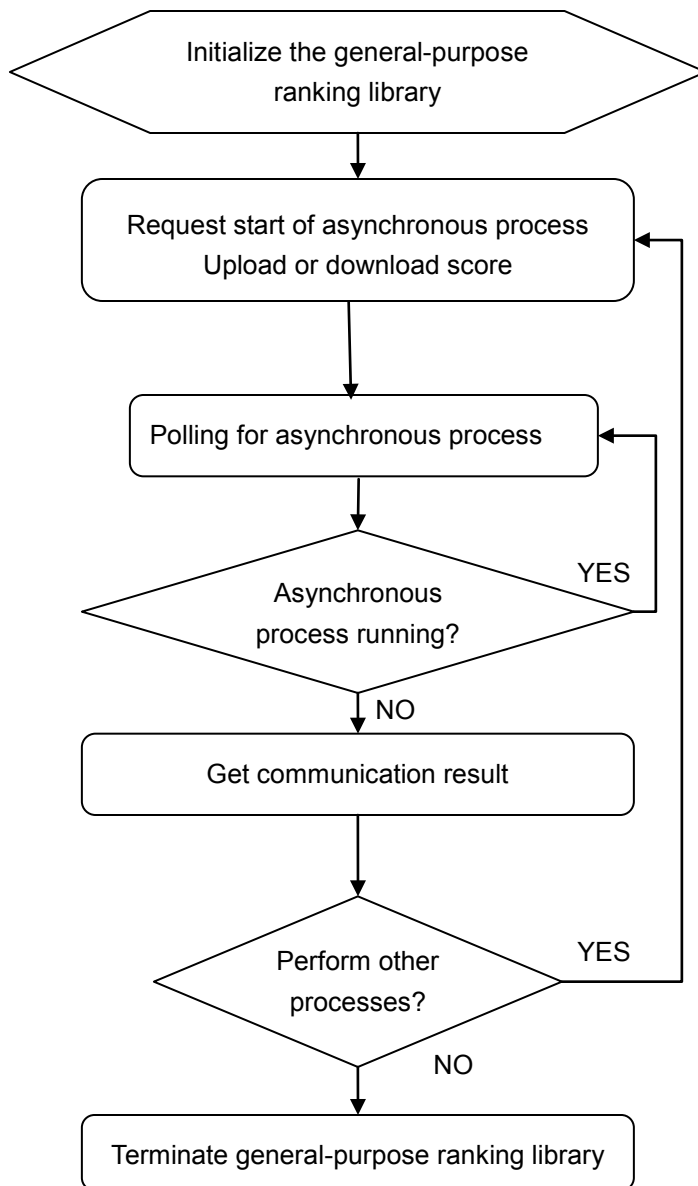
## 2 How to Use the Library

This chapter explains how to use the general-purpose ranking library.

### 2.1 Process Flow

The process flow for the general-purpose ranking library is shown in Figure 2-1.

**Figure 2-1 General-Purpose Ranking Library Process Flow**



**Note:** The general-purpose ranking library must be terminated when an asynchronous process is cancelled or when an error is generated.

### 2.1.1 Initializing the General-Purpose Ranking Library

---

The first step involves calling the `DWC_RnkInitialize` function to initialize the general-purpose ranking library. The arguments of this function take the following data.

- Initialization data

Specifies the initialization data string for the general-purpose ranking library that is disclosed by Nintendo. Every title is assigned its own unique initialization data, so you should manage this data with great care.

- User data

Specifies the `DWCUserData` structure for the user data. The account data must be for an authenticated account. Initialization will fail if it is an unauthenticated account that has never been connected to Nintendo Wi-Fi Connection.

The general-purpose ranking server has been prepared in two versions: one for development and one for production. Which server is used as the connection target is determined at the time of initialization. If the `DWC_InitForProduction` function is used for initialization, the connection target is the production server. If the `DWC_InitForDevelopment` function is used, the target is the development server.

The development server and the production server have independent databases, so the ranking in each can be constructed differently.

Be sure to use the production server for your commercial product.

You can connect to the general-purpose ranking development server even when DWC was initialized using the `DWC_InitForProduction` function by configuring the DNS server to 125.206.241.210.

#### Code 2-1 Initialization of the Ranking Library

```
DWCRnkError res;

// Initialize the Ranking library
res = DWC_RnkInitialize( RANKING_INITDATA,
                        &userdata );

if ( res != DWC_RNK_SUCCESS ){

    break; // Error process on failure to initialize

}
```

### 2.1.2 Uploading Scores

---

Scores are uploaded using an asynchronous process. Call the `DWC_RnkPutScoreAsync` function to begin this process. Only one asynchronous process can be executed at a time.

Polling is conducted from the start of the asynchronous process until the process is completed by the `DWC_RnkProcess` function (described below).

You can specify the category ID, region, score, and user-defined data for a score that you are uploading.

The category ID is an identifier used because a game title can have multiple ranking lists. For the category ID you can specify any value between 0 and 1000 (`DWC_RNK_CATEGORY_MAX`).

The region can be specified as `DWC_RNK_REGION_JP`, `DWC_RNK_REGION_US`, `DWC_RNK_REGION_EU`, `DWC_RNK_REGION_KR`, `DWC_RNK_REGION_CN`, or `DWC_RNK_REGION_ALL` (for Japan, North America, Europe, Korea, China, and all regions, respectively). The region specified here can be used as a filter when downloading scores. You could, for example, get one ranking for Japan and another ranking for both Japan and North America combined.

The user-defined data can take any binary data up to a maximum of 764 bytes (`DWC_RNK_DATA_MAX`).

If the user's own score is already registered with the same category ID, a newly uploaded score will always overwrite the existing score.

#### Code 2-2 Request to Start Uploading Scores

```
DWCRnkError res;

// Request to start uploading scores
res = DWC_RnkPutScoreAsync(      10,                // Category
                               DWC_RNK_REGION_JP,    // Region
                               1234,                // Score
                               (void*)"test data",    // User-defined data
                               strlen("test data") + 1 );

if( res != DWC_RNK_SUCCESS ){

    break; // Error process on failure

}
```

### 2.1.3 Downloading Scores

Scores are downloaded using an asynchronous process. Call the `DWC_RnkGetScoreAsync` function to begin this process. Only one asynchronous process can be executed at a time.

Polling is conducted from the start of the asynchronous process until the process is completed by the `DWC_RnkProcess` function (described below).

There are four different Get modes for score downloading. Each mode gets a different set of data.

Table 2-1 shows these four modes and what each mode gets.

**Table 2-1 Get Modes for Downloading Scores**

Get Mode	Data Retrieved
Rank order	Gets the user's own place in the ranking. Ranking is determined by comparing registered scores. A parameter specifies whether to rank in ascending or descending order.
Top-rankings list	Gets a list of rankings, starting with the top score and continuing for the specified number of scores.
Near-rankings list	Gets a list of scores ranked in order near your score. The list contains the specified number of scores.
Friend-rankings list	Gets a list of scores for as many as 64 friends, ranked in order of score.

**2.1.3.1 How to Specify the DWCRnkGetParam Structure**

This section explains how to specify the parameters of the `DWCRnkGetParam` structure, which is passed as an argument to the `DWC_RnkGetScoreAsync` function.

`DWCRnkGetParam` is defined as a structure that internalizes multiple unions, so the parameters must be specified in the appropriate fields for the different Get modes.

**The DWCRnkGetParam.size Field**

This parameter, which specifies the size of the structure, is common to all Get modes. The size of the structure for each Get mode is shown in Table 2-2.

**Table 2-2 Size Specified in Parameter for Each Get Mode**

Get Mode	Specified Size
Rank order ( <code>DWC_RNK_GET_MODE_ORDER</code> )	<code>sizeof( DWCRnkGetParam.order )</code>
Top-rankings list ( <code>DWC_RNK_GET_MODE_TOPLIST</code> )	<code>sizeof( DWCRnkGetParam.toplist )</code>
Near-rankings list ( <code>DWC_RNK_GET_MODE_NEAR</code> ) ( <code>DWC_RNK_GET_MODE_NEAR_HI</code> ) ( <code>DWC_RNK_GET_MODE_NEAR_LOW</code> )	<code>sizeof( DWCRnkGetParam.near )</code>
Friend-rankings list ( <code>DWC_RNK_GET_MODE_FRIENDS</code> )	<code>sizeof( DWCRnkGetParam.friends )</code>

**The DWCRnkGetParam.order Field**

These parameters are specified for the Rank Order Get mode, as shown in Table 2-3.

**Table 2-3 Parameters Specified for Rank Order Get Mode**

Parameter	Definition
DWCRnkGetParam.order.sort	Specifies the sort order for the scores: DWC_RNK_ORDER_ASC: ascending DWC_RNK_ORDER_DES: descending
DWCRnkGetParam.order.since	Gets the rankings that have changed since the specified number of minutes ago. If 0 is specified, the rankings for all data are downloaded. For example, if 180 is specified, then rankings of the scores that have been updated in the past 3 hours (that is, 180 minutes) are downloaded. Even if a user's scores have not been uploaded within the specified period, the ranking within the specified period for the most recently uploaded score is obtained.

**The DWCRnkGetParam.toplist Field**

These parameters are specified for the Top-Rankings List Get mode, as shown in Table 2-4.

**Table 2-4 Parameters Specified for Top-Rankings List Get Mode**

Parameter	Definition
DWCRnkGetParam.toplist.sort	Specifies the sort order for the scores: DWC_RNK_ORDER_ASC: ascending DWC_RNK_ORDER_DES: descending
DWCRnkGetParam.toplist.since	Gets the rankings that have changed since the specified number of minutes ago. If 0 is specified, the rankings for all of the data are downloaded. For example, if 180 is specified, then rankings that have been updated in the past 3 hours (that is, 180 minutes) are downloaded.
DWCRnkGetParam.toplist.limit	Specifies the maximum number of listed rankings to get. Can take a value between 1 and 30 (DWC_RNK_GET_MAX).

**The WCRnkGetParam.near Field**

These parameters are specified for the Near-Rankings List Get mode, as shown in Table 2-5.

**Table 2-5 Parameters Specified for the Near-Rankings List Get Mode**

Parameter	Definition
DWCRnkGetParam.near.sort	Specifies the sort order for the scores: DWC_RNK_ORDER_ASC: ascending DWC_RNK_ORDER_DES: descending
DWCRnkGetParam.near.since	Gets the rankings that have changed since the specified number of minutes ago. If 0 is specified, the rankings for all of the data are downloaded. For example, if 180 is specified then rankings that have been updated in the past 3 hours (that is, 180 minutes) are downloaded.

Parameter	Definition
DWCRnkGetParam.near.limit	Specifies the maximum number of listed rankings to get. Can take a value between 2 and 30 (DWC_RNK_GET_MAX). Since user's own score is invariably at the head of the list, you must specify a value of at least 2.

### The DWCRnkGetParam.friends Field

These parameters are specified for the Friend-Rankings List Get mode, as shown in Table 2-6.

**Table 2-6 Parameters Specified for the Friend-Rankings List Get Mode**

Parameter	Definition
DWCRnkGetParam.friends.sort	Specifies the sort order for the scores: DWC_RNK_ORDER_ASC: ascending DWC_RNK_ORDER_DES: descending
DWCRnkGetParam.friends.since	Gets the rankings that have changed since the specified number of minutes ago. If 0 is specified, the rankings for all of the data are downloaded.  For example, if 180 is specified, then rankings that have been updated in the past 3 hours (that is, 180 minutes) are downloaded.
DWCRnkGetParam.friends.limit	Specifies the maximum number of listed rankings to get. Can take a value between 2 and 65 (DWC_RNK_FRIENDS_MAX). Since user's own score is invariably at the head of the list, you must specify a value of at least 2.
DWCRnkGetParam.friends.friends[64]	Specifies a list of friend GS profile IDs. The list can have up to 64 (DWC_RNK_FRIENDS_MAX) values. If the list does not have 64 values, store the entries from the start and fill the remaining region with zeroes.

### Code 2-3 Request to Start Score Download to Get Rank Order

```
DWCRnkError res;

// Start request to get rank order
DWCRnkGetParam param;           // Parameter for getting rank order
param.size = sizeof( param.order );
param.order.since = 0;
param.order.sort = DWC_RNK_ORDER_ASC;

res = DWC_RnkGetScoreAsync(      DWC_RNK_GET_MODE_ORDER, // Mode
                              10,                               // Category
                              DWC_RNK_REGION_JP,               // Region
                              &param );                       // Parameter

if( res != DWC_RNK_SUCCESS ){
```

```

    break; // Failure error process

}

```

#### Code 2-4 Request to Start Score Download to Get Top-Ranking List

```

DWC_RnkError res;

// Start request to get rank order
DWC_RnkGetParam param;          // The parameter when getting top rankings
param.size = sizeof( param.toplist );
param.toplist.since = 0;
param.toplist.sort = DWC_RNK_ORDER_ASC;
param.toplist.limit = 10;

res = DWC_RnkGetScoreAsync(      DWC_RNK_GET_MODE_TOPLIST,      // Mode
                                10,                             // Category
                                DWC_RNK_REGION_JP,              // Region
                                &param );                       // Parameter

if( res != DWC_RNK_SUCCESS ){

    break; // Failure error process

}

```

#### 2.1.3.2 Get Communication Result in Rank Order Get Mode

When the Get mode is rank order, once the process to download the score has ended normally, you can use the `DWC_RnkResGetOrder` function to get the communications result. Note that this function will fail if `DWC_RnkGetScoreAsync` was called in any other Get mode setting.

If the user's own score has not been uploaded to the server, the function returns 0.

#### 2.1.3.3 Get Communication Result in Top-, Near-, and Friend-Rankings List Get Modes

Normally, when the Get mode is set to one of the ranking list modes, you can use the `DWC_RnkResGetRowCount` function to get the number of rows in the list of score rankings once the download process has ended. You can also access the list by using the `DWC_RnkResGetRow` function to get the data for each row. (See Note 1 on the following page.)

The **order** field in the `DWC_RnkData` structure stores a rank-order value only in the row that holds the user's own score. Because 0 is stored in all the other rows in the **order** field, you will need to implement suitable numbering inside your game. (See Note 2 on the following page.) An exception to this is when the mode is set to get the top rankings list. When this is the case, 0 is stored in *all* of the rows in the **order** field, even when the user's own score is included in this list.

When multiple users have the same score, there is no defined way of ordering them in the list.

If the user's own score has not been registered to the server and an attempt is made to get a near-rankings list or a friend-rankings list, the score is treated as 0. In addition, the region code is returned as -1, and the user-defined data is empty.

When scores are downloaded in the near-rankings list Get mode or friend-rankings list Get mode, the user's own score is always stored in the first index of the list, regardless of the interval specified in `since`.

**Note 1:** The pointer to the user-defined data of the `DWCRnkData` structure obtained by the `DWC_RnkResGetRow` function (for example, the `void* userdata` member) directly references the internal communication buffer. Consequently, the contents of this buffer are lost when the general-purpose ranking library terminates or the next asynchronous process starts.

**Note 2:** When the Get mode is near-rankings list and it cannot be determined how many others with the same score rank higher than the user, it is impossible to get a correct ranking for those players who are higher up in the ranking order than the user. Depending on how the list is displayed, the user might notice the discrepancy in the list that is retrieved using the top-ranking list Get mode. Because the specification takes the load on the server into account, there is fundamentally no way to resolve this issue. Instead, you will need to implement some kind of compromise, such as displaying only the user's own ranking, broadening the score range so there is less of a chance that scores overlap, or using some form of numbering that may incorporate a certain margin of error.

#### 2.1.3.4 Get the Communication Result, Get Totals

Call the `DWC_RnkResGetTotal` function to get the total number of ranked scores for the rank-order value obtained by the `DWC_RnkGetScoreAsync` function. This total value is the total number of scores that met the filtering conditions specified when `DWC_RnkGetScoreAsync` was called.

#### 2.1.3.5 Get List of Rankings Among Rivals

In friend-rankings list Get mode, you can specify the GS profile IDs of Rivals in the GS profile ID list. This gets a list of rankings among rivals. Note that exchanges of data among users who are not registered friends must conform to the *Nintendo Wi-Fi Connection Concept Guidelines for DS*.

#### Code 2-5 Accessing the Obtained Ranking List

```
DWCRnkError res;
u32 count;

// Get the number of rows in the obtained list
res = DWC_RnkResGetRowCount( &count );

if( res != DWC_RNK_SUCCESS ){

    goto exit; // Error processing on failure

}
```

```

// Get one row at a time and output for debugging
for( i=0; i<count; i++ ){
    DWCRnkData data;

    if( DWC_RnkResGetRow( &data, (u32)i ) != DWC_RNK_SUCCESS ){
        break; // Error
    }

    OS_TPrintf("%dth score=%d pid=%d rgn=%d update=%d data=%s ¥n",
               data.order, data.score, data.pid, data.region,
               data.lastupdate, data.userdata );
}

```

#### 2.1.4 Polling the Asynchronous Process to Get the Communication Result Status

Conduct polling once the asynchronous process has started by periodically calling the `DWC_RnkProcess` function. We recommend calling the function at a frequency of around once every frame (this also works at 30 fps).

The `DWC_RnkProcess` function returns `DWC_RNK_SUCCESS` while the asynchronous process is executing. When there are no more tasks, the function returns `DWC_RNK_PROCESS_NOTASK`. By monitoring this return value, you can detect when the asynchronous process has ended.

If an error occurs during the process, the function returns `DWC_RNK_IN_ERROR`. If there is an error, subsequent processing cannot continue. Terminate the general-purpose ranking library, and try starting again from the initialization process.

If the timeout (30 seconds) is reached, the `DWC_RnkProcess` function returns `DWC_RNK_PROCESS_TIMEOUT`.

##### Code 2-6 Asynchronous Process Polling

```

// Asynchronous process
while( (res = DWC_RnkProcess()) == DWC_RNK_SUCCESS ){

    // Wait for V blank
    GameWaitVBlankIntr();

}

switch( res ){
case DWC_RNK_PROCESS_NOTASK:
    // Asynchronous processing completed
    break;

```

```
case DWC_RNK_PROCESS_TIMEOUT:
    // Timeout (30 seconds)
    break;
case DWC_RNK_IN_ERROR:
    // Error processing on failure
    goto exit;
}

// Get communication result
switch( DWC_RnkGetState() ){
case DWC_RNK_STATE_COMPLETED:
    // Successful
    break;

case DWC_RNK_STATE_ERROR:
    // Error processing on failure
    goto exit;
}
```

## 2.2 Terminating the General-Purpose Ranking Library

---

Terminate by calling the `DWC_RnkShutdown` function after all processing has completed or the process has quit because of an error. Calling this function releases the memory that was used by the general-purpose ranking library and resets its internal state.

The pointer returned by the `DWC_RnkResGetRow` function directly references the Receive buffer, so it becomes invalidated when the general-purpose ranking library terminates. If you need to save data, be sure to do so before terminating the library.

## 2.3 Canceling a Process and Error Handling

---

To cancel an asynchronous process while it is executing, call the `DWC_RnkCancelProcess` function.

When a process is canceled, the general-purpose ranking library enters into an error state. Because subsequent processing will not proceed when an error has been generated, terminate the library and try again starting from the initialization process.

Because of the nature of the Internet, a network error may lead to the generation of an error while a process is executing (even when the general-purpose ranking library is being used properly). For this reason, be sure to implement a suitable error-handling process.

## 2.4 About the rank\_sample Sample Program

---

The DWC package includes a sample program that uses the general-purpose ranking library. The game name for this sample is `dwctest`.

All developers use common initialization data with this sample. As a result, the ranking data you register on the server may be viewed and altered by other developers. Do not develop a real game using the initialization data from this sample.

## 2.5 Amount of Memory Used

---

The general-purpose ranking library uses the `DWC_Alloc` function internally to allocate memory.

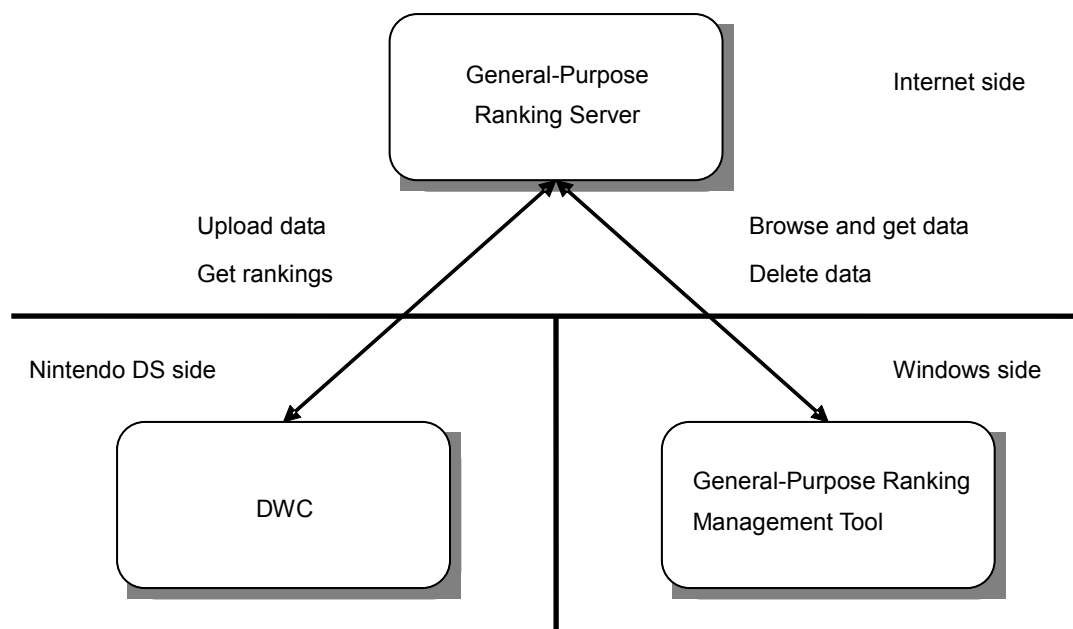
As a rule of thumb, the amount of memory used for uploading scores is 2 x (size of user-defined data + approximately 200 bytes). The amount of memory used for downloading scores is approximately 200 bytes + the total volume of score data being downloaded. This volume depends on the size of user-defined data and the maximum number of items in the obtained ranking list.

## 3 General-Purpose Ranking Management Tool

### 3.1 Overview

The General-Purpose Ranking Management Tool is a Windows application for managing data that has been uploaded to the general-purpose ranking server using the DWC's general-purpose ranking library. Use the tool to browse, get, and delete data.

**Figure 3-1 Organization**



### 3.2 File Structure

The General-Purpose Ranking Management Tool is composed of the following files.

- `DWCRankingAdmin.exe` Application executable file
- `DWCRankingAdmin.exe.config` Application settings file

### 3.3 Execution Environment

#### 3.3.1 Install .NET Framework

To use the General-Purpose Ranking Management Tool, you must have Microsoft .NET Framework version 2.0 or later installed on the computer where the tool will run. If it is not on the computer, use Windows Update or some other means to install it.

---

### 3.3.2 Install the Settings File

---

An encryption key is required for the General-Purpose Ranking Management Tool to exchange encrypted communications with the general-purpose ranking server.

The Settings file is the `admin_setting.txt` file, distributed by Nintendo. Place this file in the same folder as `DWCRankingAdmin.exe`.

---

### 3.3.3 Configure Communications

---

Because the General-Purpose Ranking Management Tool communicates using the Internet, run the tool on a computer that is connected to the Internet.

The Internet Explorer settings are used as the communications settings by the tool.

---

## 3.4 Communications Load on the General-Purpose Ranking Server

---

The General-Purpose Ranking Management Tool can place a very significant load on the general-purpose ranking server because it has the ability to gather large amounts of information at once. The people using the tool should manage projects individually. Only a small number of people should use the tool at the same time.

To reduce the load on the server, data access is limited to a maximum of 100 entries for each time data is displayed.

---

## 3.5 Time Zones of Acquired Data

---

- Time displayed under Last Modified in the management tool's list:

If **use UTC** is selected, the time is displayed in UTC.

If **use UTC** is *not* selected, the time is displayed in the local machine's time zone.

- Data acquired through Get TSV in the management tool:

If **use UTC** is selected, the time zone acquired with UTC is specified.

If **use UTC** is *not* selected, the local time zone is specified.

The time in the acquired data is specified by the server's time zone (PDT or PST).

- Web services:

The time zone acquired with UTC is specified.

The time in the acquired data is specified by the server's time zone (PDT or PST).

---

## 3.6 Starting the General-Purpose Ranking Management Tool

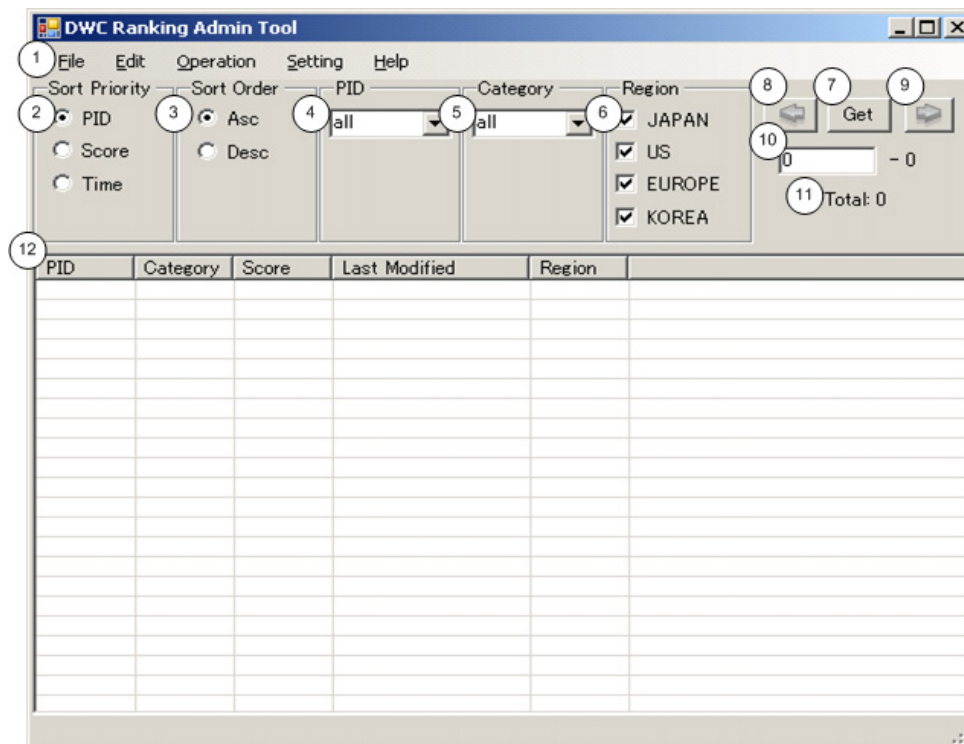
---

To start the General-Purpose Ranking Management Tool, run `DWCRankingAdmin.exe`.

### 3.7 Screen Structure

Figure 3-2 shows the layout of the main window displayed by the DWC's General-Purpose Ranking Management Tool.

**Figure 3-2 General-Purpose Ranking Management Tool Main Window**



When you start the General-Purpose Ranking Management Tool, a main window like the one in Figure 3-2 is displayed. This main window displays data and specifies conditions. The various features in the different parts of the window are explained below.

1. This is the application menu. The various features of this menu are explained in section 3.8 Menus.
2. Specifies the way to sort the data for display.

**PID** Sorts based on the order of the GS profile IDs.

**Score** Sorts based on the order of the scores.

**Time** Sorts based on the order of the latest update.

3. Specifies the sort order.  
**Asc** Sorts in ascending order.  
**Desc** Sorts in descending order.
4. Specifies the PID for the data to display. Enter a decimal number or specify **all** to show data for all PIDs.
5. Specifies the category ID for the data to display. Enter a decimal number or specify **all** to show data for all category IDs.
6. Specifies the region(s) for the data to display. Each region can be separately turned on and off.  
**JAPAN** Japan  
**US** North America  
**EUROPE** Europe  
**KOREA** Korea  
**CHINA** China
7. Click this button to display a data list according to the conditions that have been specified.
8. The **previous** button. This button is enabled when data has been displayed using an offset greater than 0. Click this button to display data using the offset that results from subtracting 100 from the current offset value.
9. The **next** button. This button is enabled when there is additional data. Click this button to display data using the offset that results from adding 100 to the current offset value.
10. Use this text box to enter an offset value for the display of data. The offset serves as the starting point for the display of 100 sets of data, starting in order from the specified offset value.
11. Shows the number of sets of data that meet the specified conditions. The value shown here gets updated when data is displayed.
12. The data list. The data retrieved is shown in the form of a list.

## 3.8 Menus

---

This section describes the various menus available in the application.

### 3.8.1 File Menu

---

<b>Exit</b>	Exits the application.
-------------	------------------------

### 3.8.2 Edit Menu

---

<b>Copy</b>	Copies selected data from the data list to the clipboard.
-------------	---

<b>Select All</b>	Selects all data on display in the data list.
-------------------	---

### 3.8.3 Operation Menu

<b>get TSV</b>	Opens the <b>Get TSV</b> dialog box to get a TSV file. See section 3.9.1 Get TSV Dialog Box.
<b>get UserData</b>	Opens the <b>Get UserData</b> dialog box to get the user-defined data. See section 3.9.2 Get UserData Dialog Box.
<b>Delete Entry</b>	Opens the <b>Delete Entry</b> dialog box to delete data. See section 3.9.3 Delete Entry Dialog Box.

### 3.8.4 Settings Menu

<b>use UTC</b>	When enabled, time is shown using UTC (Coordinated Universal Time). See section 3.5 Time Zones of Acquired Data.
<b>Proxy Setting</b>	Opens the Proxy Server Settings dialog box to configure the proxy server. See section 3.9.4 Proxy Server Settings Dialog Box.
<b>Public Server</b>	When enabled, the connection is made to the production general-purpose ranking server.

### 3.8.5 Help Menu

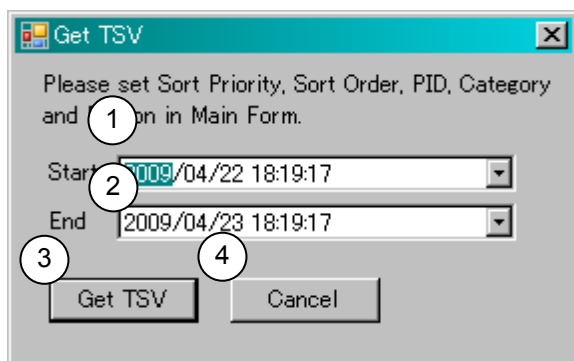
<b>Version</b>	Displays version information.
----------------	-------------------------------

## 3.9 Dialog Boxes

### 3.9.1 Get TSV Dialog Box

Use the **Get TSV** dialog box to get data in TSV format. The following conditions for getting a TSV file are used as specified in the main window: the sort order, PID, category ID, and region

**Figure 3-3 Get TSV Dialog Box**



The **Get TSV** dialog box also has these additional features (see Figure 3-3).

1. Gets data starting from the date and time specified.

2. Gets data up until the date and time specified.
3. Starts the process of getting the TSV file.
4. Cancels the process. The dialog box closes, and no TSV file is retrieved.

The **Save File** dialog box opens when the process of getting the file has completed. Specify in this dialog box where to save the file.

The retrieved TSV file has one entry per line. Each entry has the following format.

```
100000 (tab) 10 (tab) 4 (tab) Thu Apr 13 00:38:41 PDT 2006 (tab) 5758 (tab)
dGVzdCBkYXRhAA==
```

The tabs serve as delimiters. From left to right, the data elements included in the entry are the PID, the category ID, the region code, the time of the last update, the score, and the user-defined data (base64 encoded). Note that in the base64 encoding used in this TSV file, “-” is replaced with “\_” and “+” with “/”.

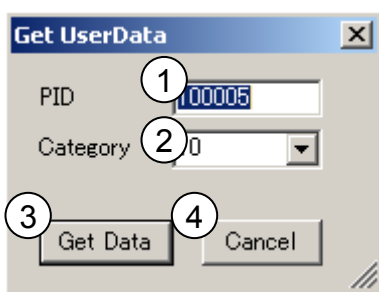
Up to 5,000 entries can be retrieved in a single TSV file. A warning is displayed if this limit has been exceeded. If this occurs, change the specified conditions to reduce the number of entries in the file (such as specifying a shorter time period) and then retrieve the TSV file again.

It can take longer than a minute to retrieve the TSV file if the file contains a large number of entries. Because retrieving the TSV file is a processing-intensive task, it should only be done once every 30 minutes or so.

### 3.9.2 Get UserData Dialog Box

Use the **Get User Data** dialog box to get user-defined data. The data is saved to a file in binary format.

Figure 3-4 Get UserData Dialog Box



This dialog box has the following parts (see Figure 3-4).

1. Specifies the PID for the data to get. If data has been selected from the data list, the PID of the selected data is already filled in when the **Get UserData** dialog box opens.
2. Specifies the category ID for the data to get. If data have been selected from the data list, the category ID of the selected data is already filled in when the **Get UserData** dialog box opens. Note that **all** cannot be specified.

3. Starts the process of getting the user-defined data file.
4. Cancels the process of getting the file and returns to the main window.

The **Save File** dialog box opens when the process of getting the file has completed. Specify where to save the file in this dialog box.

### 3.9.3 Delete Entry Dialog Box

Use the **Delete Entry** dialog box to delete data. Note that once data has been deleted, it cannot be recovered.

**Figure 3-5 Delete Entry Dialog Box**



This dialog box has the following parts (see Figure 3-5).

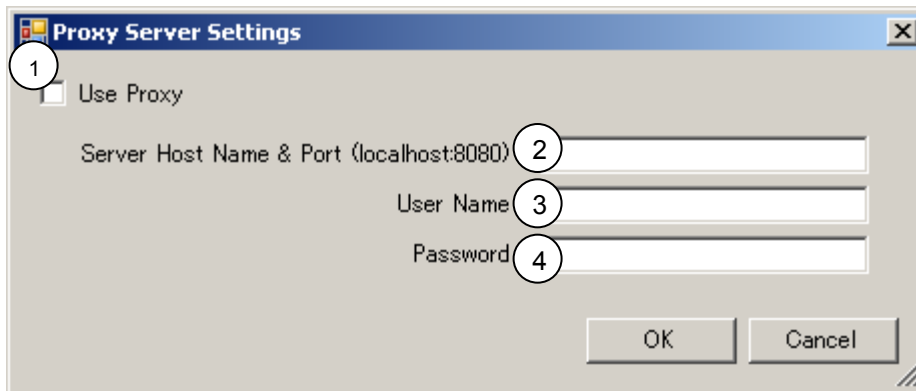
1. Specifies the PID for the data to delete. If data has been selected from the data list, the PID of the selected data is already entered in this box when the dialog box opens. In the development server, all PIDs may be removed at once by selecting **all**.
2. Specifies the category ID for the data to delete. If data has been selected from the data list, the category ID of the selected data is already entered in this box when the dialog box opens. If **all** is specified, entries of all category IDs are deleted.
3. Starts the process of deleting the data.
4. Cancels the process and returns to the main window.

The data list is automatically updated when data is deleted.

**Note:** Specifications prevent you from selecting **all** for deletion when using the production server.

### 3.9.4 Proxy Server Settings Dialog Box

Figure 3-6 Proxy Server Settings Dialog Box



This dialog box configures the proxy server (see Figure 3-6).

1. If selected, the specified proxy server is used. If not selected, the Internet Explorer setting is used. If proxy authentication is required, it must be configured here because the Internet Explorer settings do not allow proxy authentication.
2. Sets the proxy server's host name and port number.
3. Sets the user name for proxy authentication. This is left blank if proxy authentication is not used.
4. Sets the password for proxy authentication. This is left blank if proxy authentication is not used.

**Note:** The user name and password set here are saved in the settings file for the management tool and enabled at the next startup. However, this is a security risk because the user name and password are not encrypted in the settings file. Corrective measures will be taken in the next version.

## 3.10 Shortcut Keys

The following shortcut keys can be used with the General-Purpose Ranking Management Tool.

- CTRL+C Copies the selected data from the data list to the clipboard.
- CTRL+A Selects all data displayed in the data list.
- CTRL+S Opens the **Get TSV** dialog box to get a TSV file.
- CTRL+U Opens the **Get UserData** dialog box to get user-defined data.
- CTRL+D Opens the **Delete Entry** dialog box to delete data.

## 4 Web Service Development

### 4.1 About Web Services

---

The general-purpose ranking library has an interface for retrieving collected data via the Internet. This feature can be used to provide various services, such as to display rankings on a game's homepage.

### 4.2 How to Use

---

This section describes how to use this feature.

#### 4.2.1 How to Access

---

Access the addresses shown below to get general-purpose ranking data through the Internet. A tab-delimited file can be obtained as the HTTP response.

- To access the production server:

```
http://gamestats2.gs.nintendowifi.net/[gamename]/web/admin/getcsv.asp
```

- To access the development server:

```
http://sdkdev.gamespy.com/games/[gamename]/web/admin/getcsv.asp
```

**Note:** The *[gamename]* parameter takes the unique string assigned to each game title. It is included in the design statement returned from Nintendo.

#### 4.2.2 About Security

---

For security reasons, there are restrictions on the IP addresses allowed to access the general-purpose ranking Web service. If you are going use the Web service, be sure to contact [support@noa.com](mailto:support@noa.com) to inform Nintendo of the global IP address from which you will access the service.

There are no restrictions on access to the development server.

#### 4.2.3 About the Data Format

---

Data obtained from the general-purpose ranking Web service is in the format of a tab-delimited file, with one entry displayed on each line.

In order, the data elements included in each entry are the GS profile ID, the category ID, the region, the time of the last update, the score, and the user-defined data (base64-encoded). Please note that in the base64 encoding used in this file, “-” is replaced with “\_” and “+” with “/”.

#### 4.2.4 About Server Loads

---

Use of the Web service places a very large load on the general-purpose ranking server. For this reason, only get data at a frequency of around once every 30 minutes, set an appropriate value for the *since* parameter when getting data, and avoid getting the same data again.

## 4.2.5 Get Parameters

When requesting data from the general-purpose ranking server, parameters can be specified that define how to get the data. These get parameters are attached to the end of the URL in the form of a string, as shown in the example below.

```
http://.../getcsv.asp?[parameter name 1]=[value 1]&[parameter name 2]=[value 2]&...
```

Below is a list of the parameters and values that can be specified. Not all of the parameters must be specified, as when a parameter is not specified its default value is used.

<code>sort</code>	Specifies how to sort the data. The default is 0.  0 The obtained data is sorted in ascending order based on score. 1 The obtained data is sorted in descending order based on score.
<code>region</code>	Specifies which region(s) of data to get data for. The values are OR'ed together. The default value is 255 (all regions)  1 Japan 2 North America 4 Europe 8 Korea 16 China
<code>pid</code>	Specifies the PID for the data to be retrieved. It takes a numeric value. The default is to get the data for all PIDs.
<code>category</code>	Specifies the category ID for the data to be retrieved. It takes a numeric value. The default is to get the data for all category IDs.
<code>limit</code>	Specifies the maximum number of data sets to retrieve. It takes a numeric value between 1 and 5000. The default is 100.
<code>since</code>	When specified, only data updated since the specified date and time is retrieved.  The default is to retrieve everything without any date/time limit.  The parameter follows this format: year-month-day-hour-minute-second.  The parameter should be specified using UTC (Coordinated Universal Time)

Below is an example of how the URL would be specified to get a maximum of 1000 sets of data updated since 13:00 on April 1, 2006, with the category ID of 10. Do not place a zero before single-digit days and months.

```
http://.../getcsv.asp?category=10&limit=1000&since=2006-4-1-13-0-0
```

All company and product names in this document are the trademarks or registered trademarks of their respective companies.

© 2006-2009 Nintendo

The contents of this document cannot be duplicated, copied, reprinted, transferred, distributed, or loaned in whole or in part without the prior approval of Nintendo.