

# Nintendo Wi-Fi Connection

NITRO-DWC プログラミングマニュアル

— 汎用ランキング編 —

Ver 1.1.0

任天堂株式会社発行

このドキュメントの内容は、機密情報であるため、厳重な取り扱い、管理を行ってください。

## 目次

|         |  |    |
|---------|--|----|
| 1       | はじめに .....                             | 5  |
| 2       | 汎用ランキングライブラリの使用方法 .....                | 6  |
| 2.1     | 処理の流れ .....                            | 6  |
| 2.1.1   | 汎用ランキングライブラリの初期化 .....                 | 7  |
| 2.1.2   | スコアのアップロード .....                       | 8  |
| 2.1.3   | スコアのダウンロード .....                       | 9  |
| 2.1.3.1 | DWCRnkGetParam 構造体の指定方法 .....          | 9  |
| 2.1.3.2 | 通信結果の取得 順位取得モード .....                  | 13 |
| 2.1.3.3 | 通信結果の取得 ランキングリスト取得モード(トップ、周辺、友達) ..... | 13 |
| 2.1.3.4 | 通信結果の取得 母数の取得 .....                    | 13 |
| 2.1.3.5 | ライバル内のランキングリストの取得 .....                | 13 |
| 2.1.4   | 非同期処理ポーリング 通信結果状態の取得 .....             | 15 |
| 2.2     | 汎用ランキングライブラリの終了 .....                  | 16 |
| 2.3     | 処理の中断とエラー処理 .....                      | 16 |
| 2.4     | エラーコードについて .....                       | 16 |
| 2.5     | サンプル(rank_sample)について .....            | 16 |
| 2.6     | DWC_GHTTP への依存について .....               | 16 |
| 2.7     | メモリ使用量について .....                       | 16 |
| 3       | 汎用ランキング管理ツール .....                     | 17 |
| 3.1     | 概要 .....                               | 17 |
| 3.2     | ファイル構成 .....                           | 17 |
| 3.3     | 実行環境 .....                             | 17 |
| 3.3.1   | .NET Framework のインストール .....           | 17 |
| 3.3.2   | 設定ファイルのインストール .....                    | 18 |
| 3.3.3   | 通信設定 .....                             | 18 |
| 3.4     | ランキングサーバへの通信負荷について .....               | 18 |
| 3.5     | 取得するデータのタイムゾーンについて .....               | 18 |
| 3.6     | 汎用ランキング管理ツールの起動 .....                  | 18 |
| 3.7     | 画面構成 .....                             | 19 |
| 3.8     | メニュー構成 .....                           | 20 |
| 3.8.1   | File メニュー .....                        | 20 |
| 3.8.2   | Edit メニュー .....                        | 20 |
| 3.8.3   | Operation メニュー .....                   | 20 |
| 3.8.4   | Setting メニュー .....                     | 20 |
| 3.8.5   | Help メニュー .....                        | 21 |
| 3.9     | ダイアログボックス .....                        | 21 |
| 3.9.1   | Get CSV ダイアログボックス .....                | 21 |
| 3.9.2   | Get UserData ダイアログボックス .....           | 22 |

|       |                                      |    |
|-------|--------------------------------------|----|
| 3.9.3 | Delete Entry ダイアログボックス.....          | 22 |
| 3.9.4 | Proxy Server Settings ダイアログボックス..... | 23 |
| 3.10  | ショートカットキー .....                      | 23 |
| 4     | Web サービス開発 .....                     | 24 |
| 4.1   | Web サービスについて .....                   | 24 |
| 4.2   | 使い方.....                             | 24 |
| 4.2.1 | アクセス方法 .....                         | 24 |
| 4.2.2 | セキュリティについて.....                      | 24 |
| 4.2.3 | データの形式について .....                     | 24 |
| 4.2.4 | サーバへの負荷について.....                     | 24 |
| 4.2.5 | 取得パラメータについて .....                    | 25 |

## 図

|       |                              |    |
|-------|------------------------------|----|
| 図 2-1 | 処理の流れ .....                  | 6  |
| 図 3-1 | 構成図 .....                    | 17 |
| 図 3-2 | メインフォーム.....                 | 19 |
| 図 3-3 | Get CSV ダイアログボックス .....      | 21 |
| 図 3-4 | Get UserData ダイアログボックス ..... | 22 |
| 図 3-5 | Delete Entry ダイアログボックス ..... | 22 |

## 表

|       |                                 |    |
|-------|---------------------------------|----|
| 表 2-1 | スコアのダウンロードにおける取得モード.....        | 9  |
| 表 2-2 | 取得モードごとのパラメータに指定するサイズ.....      | 9  |
| 表 2-3 | 順位取得モードで設定するパラメータ .....         | 10 |
| 表 2-4 | トップランキングリストモードで設定するパラメータ .....  | 10 |
| 表 2-5 | 周辺ランキングリスト取得モードで設定するパラメータ ..... | 11 |
| 表 2-6 | 友達ランキングリスト取得モードで設定するパラメータ ..... | 11 |

## コード

|         |   |    |
|---------|---|----|
| コード 2-1 | ランキングライブラリ初期化 .....                     | 7  |
| コード 2-2 | スコアアップロードの開始リクエスト .....                 | 8  |
| コード 2-3 | スコアダウンロード(順位の取得)の開始リクエスト .....          | 12 |
| コード 2-4 | スコアダウンロード(トップランキングリストの取得)の開始リクエスト ..... | 12 |
| コード 2-5 | 取得したランキングリストへのアクセス.....                 | 14 |
| コード 2-6 | 非同期処理ポーリング .....                        | 15 |

## 改訂履歴

| 版     | 改訂日       | 改訂内容   | 承認者 | 担当者 |
|-------|-----------|--|-----|-----|
| 1.1.0 | 2007-7-19 | 2.1.2 カテゴリ数の最大値を 100 から 1000 に変更<br>2.1.3.1 一度に取得できる数を 10 から 30 に変更<br>2.1.3.1 上位/下位周辺ランキング取得に関する記述を追加<br>3.5 管理ツールのタイムゾーンに関する記述を追加<br>3.8.4/3.9.4 管理ツールのプロキシ設定に関する記述を追加 |     |     |
| 1.0.9 | 2007-4-27 | 韓国リージョンに対応   |     |     |
| 1.0.8 | 2007-4-18 | 2.1.2 ユーザー定義データのサイズを 764 バイトに変更  |     |     |
| 1.0.7 | 2007-4-10 | 表 2-3 since 指定における自分の順位取得への影響を追記<br>2.1.3.3 since 指定における自分のスコア取得への影響を追記  |     |     |
| 1.0.6 | 2007-2-2  | 3.8.1 base64 エンコードに関する誤記修正。<br>4.2.3 base64 エンコードに関する誤記修正。   |     |     |
| 1.0.5 | 2006-9-20 | 3.8.1 base64 エンコードが独自形式であることについて追記。<br>3.8.3 開発用サーバで全ての PID について削除を行うことができることについて追記。<br>4.2.1 CSV を取得する URL を変更。<br>4.2.3 base64 エンコードが独自形式であることについて追記。                  |     |     |
| 1.0.4 | 2006-7-26 | 4.2.3 CSV ファイルの形式に関する記述を訂正。  |     |     |
| 1.0.3 | 2006-7-21 | 2.1.3.3 「サーバー」→「サーバ」<br>2.1.3.3 順位を特定できないケースに関する注意書きを修正。   |     |     |
| 1.0.2 | 2006-6-19 | 2.1.3.3 順位を特定できないケースに関する注意書きを追加。   |     |     |
| 1.0.1 | 2006-6-12 | 3.6 「画面構成」の誤記修正。<br>3.8.3 「Delete Entry ダイアログボックス」の誤記修正。   |     |     |
| 1.0.0 | 2006-6-6  | 初版   |     |     |

# 1 はじめに

本ドキュメントは、NITRO-DWC(以下 DWC)の汎用ランキングライブラリの使用方法、管理ツールの使用方法、及び Web 開発インターフェースについて解説するものです。

汎用ランキングライブラリでは以下の機能が提供されます。

- ・ スコアのアップロード
- ・ 自分の順位の取得
- ・ トップランキングリストの取得
- ・ 自分に近いスコアのランキングリスト取得
- ・ 友達内、ライバル内でのランキングリストの取得

汎用ランキングライブラリは、これらの機能を実現するためのランキングサーバとの通信を行うための API を提供します。

汎用ランキングライブラリは、内部で DWC\_GHTTP を使ってランキングサーバとの通信を行います。従って、汎用ランキングライブラリと DWC\_GHTTP を同時に使うことはできません。また、汎用ランキングにおける通信エラーはすべて DWC\_GHTTP で発生するため、エラーコードは DWC\_GHTTP に依存しています。

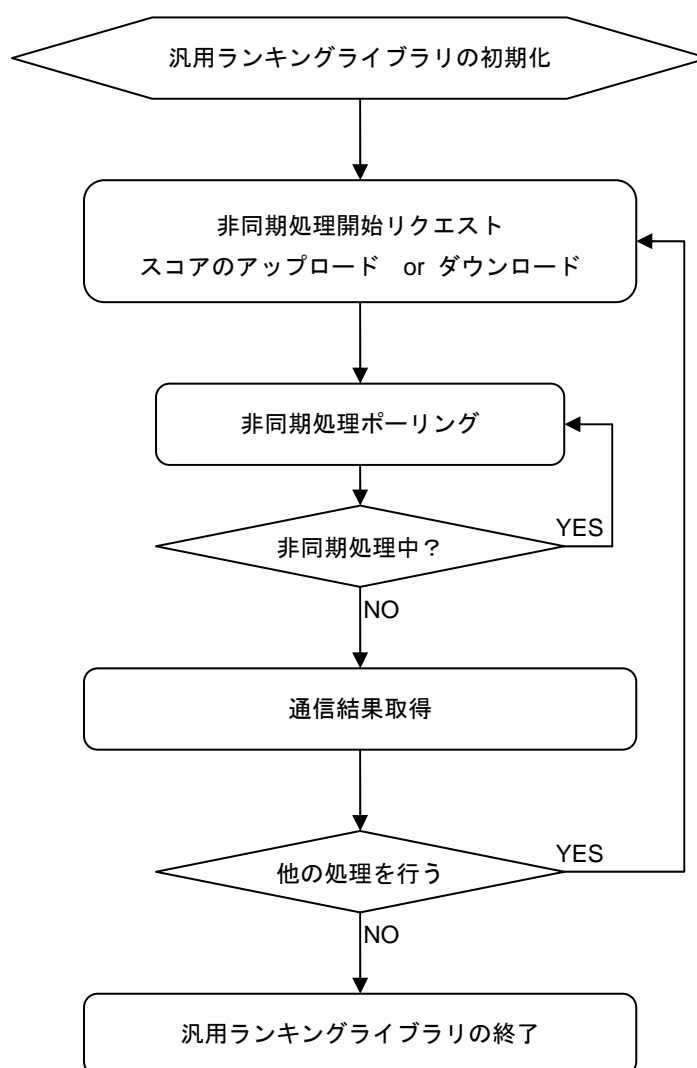
実際の使用においては、「Nintendo Wi-Fi Connection プログラミングガイドライン」に従ってください。

## 2 汎用ランキングライブラリの実用方法

この章では汎用ランキングライブラリの実用方法について解説します。

### 2.1 処理の流れ

汎用ランキングライブラリの処理の流れを以下に示します。



※非同期処理をキャンセルしたとき、またはエラーが発生したときには、  
一度汎用ランキングライブラリを終了する必要があります。

図 2-1 処理の流れ

### 2.1.1 汎用ランキングライブラリの初期化

はじめに、DWC\_RnkInitialize 関数を呼び出して汎用ランキングライブラリを初期化します。この関数の引数として渡すデータは以下のとおりです。

- ・ 初期化データ

デザインステートメント提出後に弊社より通知される、汎用ランキングライブラリの初期化データの文字列を指定します。この初期化データは、タイトルごとに固有に与えられるもので、厳重に管理してください。

- ・ ユーザーデータ

ユーザーデータの DWCUserData 構造体を指定します。このアカウントデータは認証済みのアカウントである必要があります。未だ一度も Nintendo Wi-Fi Connection に接続したことのない未認証のアカウントは初期化に失敗します。

汎用ランキングのサーバは開発向けサーバと、製品向けサーバの二つが用意されています。どちらのサーバを接続先として使用するかは、この初期化の時点で、どちらの認証サーバにログインしているかで決定されます。DWC\_SetAuthServer 関数で、DWC\_CONNECTINET\_AUTH\_RELEASE が指定されている場合は製品向けサーバに、デフォルト状態か DWC\_CONNECTINET\_AUTH\_TEST を指定した場合は開発向けサーバに接続します。

開発向けサーバと製品向けサーバはデータベースが互いに独立しているため、双方のランキングは異なるものとして構成されます。

製品では必ず製品向けサーバが使用されるようにしてください。

```
DWCRnkError res;

// ランキングライブラリの初期化
res = DWC_RnkInitialize(      RANKING_INITDATA,
                             &userdata );

if( res != DWC_RNK_SUCCESS ){

    break; // 初期化失敗 エラー処理

}
```

コード2-1 ランキングライブラリ初期化

### 2.1.2 スコアのアップロード

スコアのアップロードは非同期処理で行います。アップロードを行う非同期処理を開始するためには `DWC_RnkPutScoreAsync` 関数を呼び出します。同時に実行できる非同期処理は一つまでです。

非同期処理を開始したら後述の `DWC_RnkProcess` 関数で処理が完了するまでポーリングを行います。

アップロードするスコアには、カテゴリーID、リージョン、スコア、ユーザー定義データを指定することができます。

カテゴリーID は、一つのタイトルで複数のランキングリストを持つための識別子で、0～1000(`DWC_RNK_CATEGORY_MAX`)までを指定することができます。

リージョン は、`DWC_RNK_REGION_JP`(日本)、`DWC_RNK_REGION_US`(米国)、`DWC_RNK_REGION_EU`(欧州)、`DWC_RNK_REGION_KR`(韓国)、`DWC_RNK_REGION_ALL`(全地域)のいずれかを指定することができます。ここで指定したリージョンは、スコアをダウンロードする際にフィルターするために使うことができます。例えば、日本内でのランキングや、日本と米国を合わせたランキングを取得したりすることができます。

ユーザー定義データには、最大 764bytes (`DWC_RNK_DATA_MAX`)の任意のバイナリデータを指定することができます。

同一のカテゴリーID に既に自分のスコアが登録されている場合は、常に新しくアップロードされたスコアが上書きされます。

```
DWCRnkError res;

// スコアアップロードリクエスト開始
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; // 失敗 エラー処理

}
```

コード 2-2 スコアアップロードの開始リクエスト



### 2.1.3 スコアのダウンロード

スコアのダウンロードは非同期処理で行います。ダウンロードを行う非同期処理を開始するためには `DWC_RnkGetScoreAsync` 関数を呼び出します。同時に実行できる非同期処理は一つまでです。

非同期処理を開始したら後述の `DWC_RnkProcess` 関数で処理が完了するまでポーリングを行います。

スコアのダウンロードには、順位取得モード、トップランキングリスト取得モード、周辺ランキングリスト取得モード、友達ランキングリスト取得モードの四つの取得モードがあります。

|                      |  |
|----------------------|--|
| 順位取得モード              | 自分の順位を取得します。順位は登録されているスコアの比較によって行われ、昇順・降順をパラメータで指定します。 |
| トップランキングリスト<br>取得モード | トップから指定数のランキングリストを取得します。                               |
| 周辺ランキングリスト取得モード      | 自分のスコアに近い順に指定数のランキングリストを取得します。                         |
| 友達ランキングリスト取得モード      | 指定した最大 64 人の友達の中でのランキングリストを取得します。                      |

表 2-1 スコアのダウンロードにおける取得モード

#### 2.1.3.1 DWCRnkGetParam 構造体の指定方法

`DWC_RnkGetScoreAsync` 関数の引数として与える `DWCRnkGetParam` 構造体のパラメータの指定方法について解説します。`DWCRnkGetParam` 構造体は、複数の共用体(union)を内包した構造体として定義されており、取得モードに応じて適切なフィールドにパラメータを設定する必要があります。

- `DWCRnkGetParam.size` フィールド

すべての取得モードで共通して構造体のサイズを指定します。取得モードごとの構造体のサイズは以下のとおりです。

|  |   |
|--|---|
| 順位取得モード<br>( <code>DWC_RNK_GET_MODE_ORDER</code> )   | <code>sizeof( DWCRnkGetParam.order )</code>   |
| トップランキングリスト取得モード<br>( <code>DWC_RNK_GET_MODE_TOPLIST</code> )  | <code>sizeof( DWCRnkGetParam.toplist )</code> |
| 周辺ランキングリスト取得モード<br>( <code>DWC_RNK_GET_MODE_NEAR</code> )<br>( <code>DWC_RNK_GET_MODE_NEAR_HI</code> )<br>( <code>DWC_RNK_GET_MODE_NEAR_LOW</code> ) | <code>sizeof( DWCRnkGetParam.near )</code>    |
| 友達ランキングリスト取得モード<br>( <code>DWC_RNK_GET_MODE_FRIENDS</code> )   | <code>sizeof( DWCRnkGetParam.friends )</code> |

表 2-2 取得モードごとのパラメータに指定するサイズ

- DWCRnkGetParam.order フィールド

順位取得モードで指定します。

|                            |  |
|----------------------------|--|
| DWCRnkGetParam.order.sort  | スコアをソートする並び順を指定します。<br><br>DWC_RNK_ORDER_ASC:昇順<br><br>DWC_RNK_ORDER_DES:降順  |
| DWCRnkGetParam.order.since | 現時点から指定分前までに更新された中でのランキングを取得します。0を指定した場合はすべてのデータ中でのランキングを取得します。(例: 180 を指定すると 180 分(3 時間)以内に更新された中でのランキングを取得します。)<br><br>指定した期間内に自分がスコアをアップロードしていない状態でも、最後にアップロードしたスコアに対する指定した期間内の順位が取得されます。 |

表 2-3 順位取得モードで設定するパラメータ

- DWCRnkGetParam.toplist フィールド

トップランキングリスト取得モードで指定します。

|                              |   |
|------------------------------|---|
| DWCRnkGetParam.toplist.sort  | スコアをソートする並び順を指定します。<br><br>DWC_RNK_ORDER_ASC:昇順<br><br>DWC_RNK_ORDER_DES:降順                                       |
| DWCRnkGetParam.toplist.since | 現時点から指定分前までに更新された中でのランキングを取得します。0を指定した場合はすべてのデータ中でのランキングを取得します。(例: 180 を指定すると 180 分(3 時間)以内に更新された中でのランキングを取得します。) |
| DWCRnkGetParam.toplist.limit | 取得するランキングリストの最大数を指定します。1～30(DWC_RNK_GET_MAX)までの数値を指定できます。   |

表 2-4 トップランキングリストモードで設定するパラメータ

- ・ DWCRnkGetParam.near フィールド

周辺ランキングリスト取得モードで指定します。

|                           |   |
|---------------------------|---|
| DWCRnkGetParam.near.sort  | スコアをソートする並び順を指定します。<br><br>DWC_RNK_ORDER_ASC:昇順<br><br>DWC_RNK_ORDER_DES:降順                                       |
| DWCRnkGetParam.near.since | 現時点から指定分前までに更新された中でのランキングを取得します。0を指定した場合はすべてのデータ中でのランキングを取得します。(例: 180 を指定すると 180 分(3 時間)以内に更新された中でのランキングを取得します。) |
| DWCRnkGetParam.near.limit | 取得するランキングリストの最大数を指定します。2～30(DWC_RNK_GET_MAX)までの数値を指定できます。必ず先頭には自分のスコアがリストされるため 2 以上を指定する必要があります。                  |

表 2-5 周辺ランキングリスト取得モードで設定するパラメータ

- ・ DWCRnkGetParam.friends フィールド

友達ランキングリスト取得モードで指定します。

|                                    |   |
|------------------------------------|---|
| DWCRnkGetParam.friends.sort        | スコアをソートする並び順を指定します。<br><br>DWC_RNK_ORDER_ASC:昇順<br><br>DWC_RNK_ORDER_DES:降順                                       |
| DWCRnkGetParam.friends.since       | 現時点から指定分前までに更新された中でのランキングを取得します。0を指定した場合はすべてのデータ中でのランキングを取得します。(例: 180 を指定すると 180 分(3 時間)以内に更新された中でのランキングを取得します。) |
| DWCRnkGetParam.friends.limit       | 取得するランキングリストの最大数を指定します。2～30(DWC_RNK_GET_MAX)までの数値を指定できます。必ず先頭には自分のスコアがリストされるため 2 以上を指定する必要があります。                  |
| DWCRnkGetParam.friends.friends[64] | 友達の GS プロファイル ID のリストを指定します。最大で 64(DWC_RNK_FRIENDS_MAX)個指定できます。64 個に満たない場合は、先頭から格納し、余った領域は 0 で埋めてください。            |

表 2-6 友達ランキングリスト取得モードで設定するパラメータ

```
DWCRnkError res;

// 順位取得リクエスト開始
DWCRnkGetParam param;          // ランキング取得時のパラメータ
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; // 失敗 エラー処理

}
```

### コード 2-3 スコアダウンロード(順位の取得)の開始リクエスト

```
DWCRnkError res;

// 順位取得リクエスト開始
DWCRnkGetParam param;          // ランキング取得時のパラメータ
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; // 失敗 エラー処理

}
```

### コード 2-4 スコアダウンロード(トップランキングリストの取得)の開始リクエスト

### 2.1.3.2 通信結果の取得 順位取得モード

順位取得モードでスコアのダウンロード処理が正常終了した場合、DWC\_RnkResGetOrder 関数を呼び出すことで通信結果を取得することができます。DWC\_RnkGetScoreAsync 関数を順位取得モード以外で呼び出した場合はこの関数は失敗します。

自分のスコアがアップロードされていない状態で取得すると 0 が返されます。

### 2.1.3.3 通信結果の取得 ランキングリスト取得モード(トップ、周辺、友達)

ランキングリスト取得モードでスコアのダウンロード処理が終了した場合、DWC\_RnkResGetRowCount 関数で取得したスコアの行数を取得して、それぞれの行に対して DWC\_RnkResGetRow 関数でデータの取得を行うことで取得したランキングリストにアクセスすることができます(注1)。

DWCRnkData 構造体の order フィールドには、自分のスコアの行に限り順位が格納されます。それ以外の行では、order フィールドに 0 が格納されるため、ゲーム側で適切にナンバリングしてください(注2)。

同一のスコアを持ったユーザーが複数いる場合、リスト中での並び順は未定義です。

自分のスコアが登録されていない状態で、周辺、友達ランキングリストを取得しようとすると、自分のスコアは 0 として扱われます。また、そのときリージョンコードは -1、ユーザー定義データは空の状態で返されます。

周辺ランキングリスト取得モード、友達ランキングリスト取得モードで取得した場合、リストの最初のインデックスには since で指定した期間に関わらず必ず自分のスコアが格納されます。

注1：DWC\_RnkResGetRow 関数で取得した DWCRnkData 構造体のユーザー定義データへのポインタ(void\* userdata メンバ)は、内部の通信バッファを直接参照しています。従って、ランキングライブラリを終了、または次の非同期処理を開始するとバッファの内容は失われます。

注2：周辺ランキングリスト取得モードで、自分の上位に同スコアのユーザーが何人いるかを把握できない状況において、自分より上位のユーザーの正確な順位を特定できないケースが存在します。このとき、表示方法によっては、トップランキングリスト取得モードで取得したリストの順位との不整合がユーザーに気づかれてしまう可能性があります。これは、サーバ負荷を考慮した上での仕様であるため、根本的な解決策はありません。自分の順位しか表示しない、スコアのレンジを広げることでスコアが重複する可能性を下げる、間違っている可能性はあるが適当にナンバリングする、等の妥協策を採ってください。

### 2.1.3.4 通信結果の取得 母数の取得

DWC\_RnkResGetTotal 関数を呼び出すことで、DWC\_RnkGetScoreAsync 関数で取得した順位に対する母数を取得することができます。ここで取得する母数とは、DWC\_RnkGetScoreAsync 関数の呼び出しで指定したフィルタリングの条件に合致したスコアの数です。

### 2.1.3.5 ライバル内のランキングリストの取得

友達ランキングリスト取得モードで指定する GS プロファイル ID のリストにはライバルの GS プロファイル ID を指定することもできます。ライバルの GS プロファイル ID リストを指定することで、ライバル内でのランキングリストを取得することができます。

ただし、友達以外のユーザーとのデータのやり取りに関しては、「Nintendo Wi-Fi Connection 企画検討用ガイドライン」に準拠してください。

```
DWCRnkError res;
u32 count;

// 取得したリストの行数を得る
res = DWC_RnkResGetRowCount( &count );

if( res != DWC_RNK_SUCCESS ){
    goto exit; // 失敗 エラー処理
}

// 一行ずつ取得し、デバッグに出力する
for( i=0; i<count; i++ ){
    DWCRnkData data;

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

    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 );
}
```

#### コード 2-5 取得したランキングリストへのアクセス

### 2.1.4 非同期処理ポーリング 通信結果状態の取得

非同期処理が開始されたら `DWC_RnkProcess` 関数を定期的呼び出してポーリングを行います。呼び出す頻度は毎ゲームフレームを目安に呼び出してください(30fpsでも動作可能)。

`DWC_RnkProcess` 関数は、非同期処理中は `DWC_RNK_SUCCESS` を返します。処理するタスクがなくなると `DWC_RNK_PROCESS_NOTASK` を返します。この返り値を監視することで非同期処理の終了を検出できます。

途中でエラーが発生した場合は、`DWC_RNK_IN_ERROR` が返されます。エラーが発生すると、以後の処理を継続することができませんので、汎用ランキングライブラリを終了した上で、初期化処理から再試行してください。

```
// 非同期処理
while( (res = DWC_RnkProcess()) == DWC_RNK_SUCCESS ){

    // v ブランク待ち
    GameWaitVBlankIntr();

}

switch( res ){
case DWC_RNK_PROCESS_NOTASK:
    // 非同期処理完了
    break;
case DWC_RNK_IN_ERROR:
    // 失敗 エラー処理
    goto exit;
}

// 通信結果取得
switch( DWC_RnkGetState() ){
case DWC_RNK_STATE_COMPLETED:
    // 成功
    break;

case DWC_RNK_STATE_ERROR:
    // 失敗 エラー処理
    goto exit;
}
```

コード 2-6 非同期処理ポーリング

## 2.2 汎用ランキングライブラリの終了

---

すべての処理が完了した場合、あるいはエラーにより終了する場合は `DWC_RnkShutdown` 関数を呼び出して終了処理を行います。この関数を呼び出すことで、汎用ランキングライブラリが使用したメモリの解放や内部状態のリセットが行われます。

`DWC_RnkResGetRow` 関数によって返されるポインタは、受信バッファを直接参照しているため、汎用ランキングライブラリを終了すると無効になります。終了前に必要に応じた退避処理を行ってください。

## 2.3 処理の中断とエラー処理

---

`DWC_RnkCancelProcess` 関数を呼び出すことで、進行中の非同期処理を中断することができます。

処理を中断した場合は、汎用ランキングライブラリの状態はエラー状態に入ります。エラーが発生すると、以後の処理を継続することができませんので、汎用ランキングライブラリを終了した上で、初期化処理から再試行してください。

インターネットの性質上、汎用ランキングライブラリを正しく使用していても、ネットワークのエラーにより処理の途中でエラーが発生する可能性があります。従って、必ず適切なエラー処理を記述してください。

## 2.4 エラーコードについて

---

汎用ランキングライブラリは、内部の通信で `DWC_GHTTP` ライブラリを使用しています。従って、通信エラーに関する `DWC` のエラーコードは `DWC_GHTTP` によってセットされます。

## 2.5 サンプル(rank\_sample)について

---

`DWC` パッケージ内に汎用ランキングライブラリを使用するサンプルプログラムが同梱されています。

このサンプルの使用する初期化データ(ゲーム名は `dwctest`)はすべてのデベロッパーで共有されます。そのため、サーバ上に登録されているランキングデータは、他のデベロッパーにより閲覧、変更される可能性があります。このサンプルの初期化データを使用した実際のゲーム本編の開発は控えてください。

## 2.6 `DWC_GHTTP` への依存について

---

汎用ランキングライブラリは、内部の通信で `DWC_GHTTP` ライブラリを使用しています。そのため、汎用ランキングライブラリの初期化、終了関数は内部で `DWC_GHTTP` の初期化、終了 API が呼び出します。従って、**汎用ランキングライブラリと `DWC_GHTTP` による通信を同時に行うことはできません。**

## 2.7 メモリ使用量について

---

汎用ランキングライブラリは、内部で `DWC_Alloc` 関数を使ってメモリを確保します。確保されるメモリの目安は、スコアアップロード時は、(ユーザー定義データのサイズ + 約 200bytes) × 2、スコアダウンロード時は、約 200bytes に加え、ダウンロードされるスコアデータの総容量になります(ユーザー定義データのサイズと取得するランキングリストの最大数に依存します)。



## 3 汎用ランキング管理ツール

### 3.1 概要

汎用ランキング管理ツールは、DWC の汎用ランキングライブラリでランキングサーバへアップロードしたデータを管理するための Windows アプリケーションです。データの閲覧、取得、削除を行うことができます。

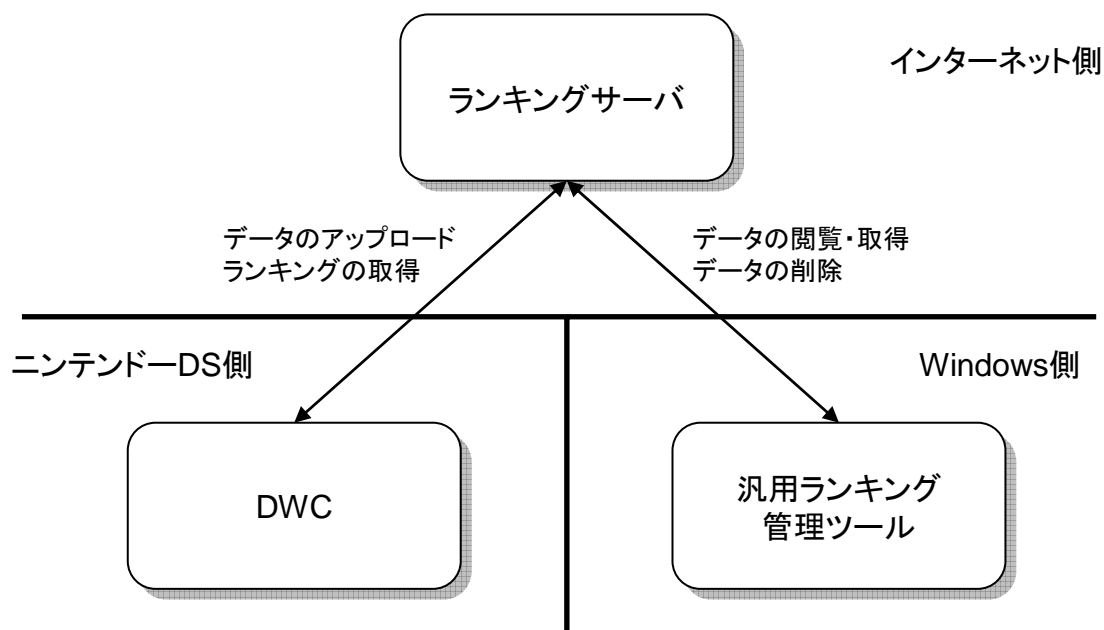


図 3-1 構成図

### 3.2 ファイル構成

汎用ランキング管理ツールは、以下のファイルから構成されます。

- ・ DWCRankingAdmin.exe … アプリケーションの実行ファイル
- ・ DWCRankingAdmin.exe.config … アプリケーションの設定が保存されるファイル

### 3.3 実行環境

#### 3.3.1 .NET Framework のインストール

汎用ランキング管理ツールを使用するには、実行するコンピュータに .NET Framework 2.0 以上がインストールされている必要があります。インストールされていない場合は、Windows Update などからインストールしてください。

### 3.3.2 設定ファイルのインストール

---

汎用ランキング管理ツールは、ランキングサーバと暗号化された通信を行うため、暗号鍵を必要とします。

設定ファイルは、弊社担当窓口から配布される `admin_setting.txt` というファイルです。`admin_setting.txt` は、`DWCRankingAdmin.exe` と同じフォルダに置いてください。

### 3.3.3 通信設定

---

汎用ランキング管理ツールはインターネットと通信しますので、インターネットに接続したコンピュータで実行してください。

通信設定は、Internet Explorer の設定を使用します。

## 3.4 ランキングサーバへの通信負荷について

---

汎用ランキング管理ツールは、一度に大量の情報を取得することができるため、ランキングサーバへ大きな負荷をかけます。ツールを使用する人はプロジェクトごとに管理し、同時に使用するのは数人程度としてください。

また、負荷軽減のため、データは表示を行うたびに最大 100 件ずつ取得するようになっています。

## 3.5 取得するデータのタイムゾーンについて

---

- ・ 管理ツールのリスト中の「Last Modified」に表示される時間
  - 「use UTC」にチェック：UTC で表示されます
  - 「use UTC」にチェックなし：ローカルマシンのタイムゾーンで表示されます
- ・ 管理ツールの Get CSV で取得する場合
  - 「use UTC」にチェック：UTC で取得する時間帯を設定します
  - 「use UTC」にチェックなし：ローカルタイムゾーンで取得する時間帯を設定します取得されるデータ中の時間はサーバのタイムゾーンが指定されています(PDT か PST)
- ・ Web サービス
  - UTC で取得する時間帯を設定します取得されるデータ中の時間はサーバのタイムゾーンが指定されています(PDT か PST)

## 3.6 汎用ランキング管理ツールの起動

---

汎用ランキング管理ツールは、`DWCRankingAdmin.exe` を実行することで起動します。

### 3.7 画面構成

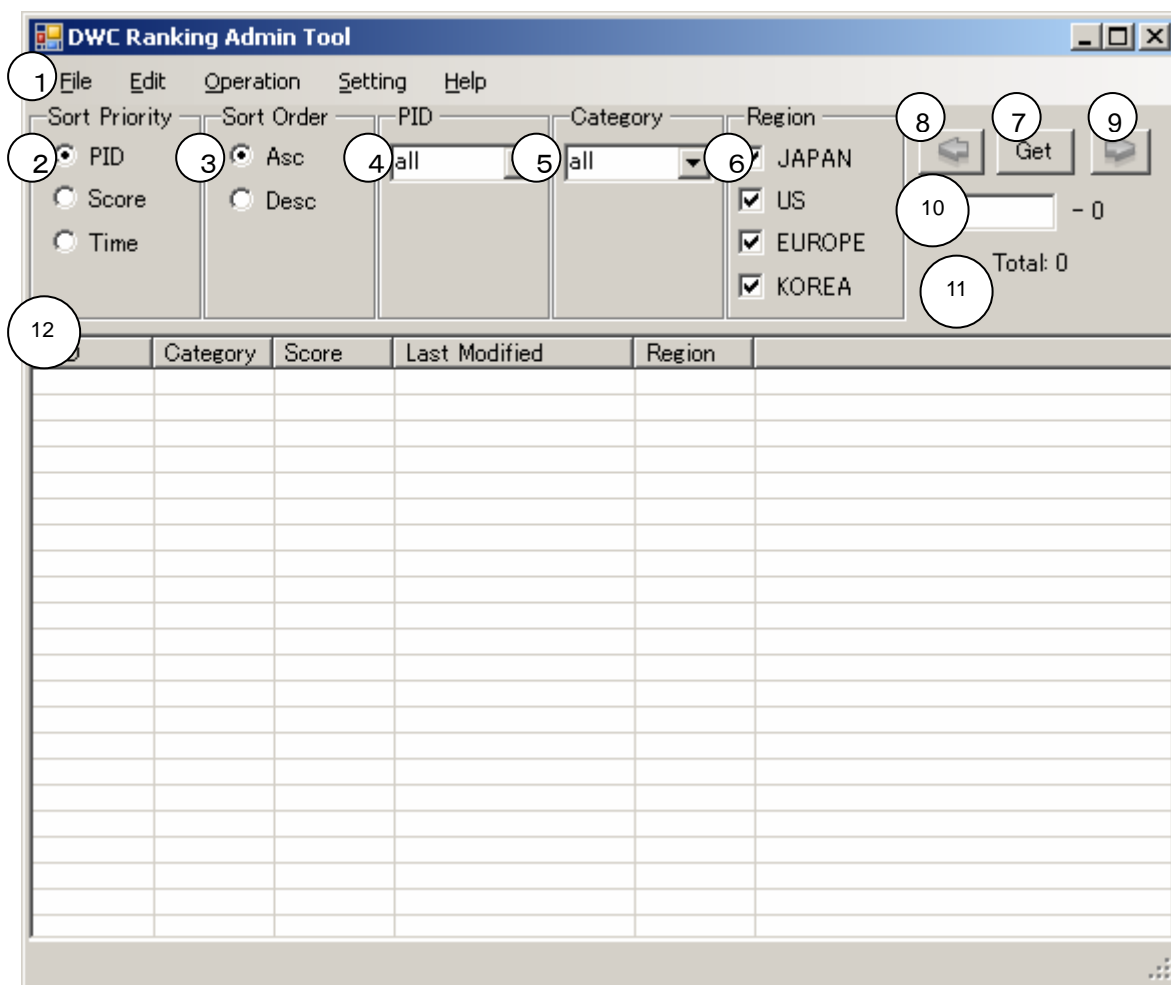


図 3-2 メインフォーム

汎用ランキング管理ツールを起動すると、上のようなアプリケーションのメインフォームが表示されます。メインフォームでは、データの表示と条件の指定を行います。それぞれの場所の機能は以下の通りです。

- ① アプリケーションのメニューです。メニューの機能は後述します。
- ② 表示するデータのソート順を指定します。いずれかを選ぶことができます。
  - ・ PID … GS プロファイル ID の順番でソートします。
  - ・ Score … スコアの順番でソートします。
  - ・ Time … 最終更新時刻の順番でソートします。
- ③ ソートの昇順・降順を指定します。いずれかを選ぶことができます。
  - ・ Asc … データを昇順(ascending order)でソートします。
  - ・ Desc … データを降順(descending order)でソートします。
- ④ 表示するデータの PID を指定します。十進数の数字、もしくは"all"を入力することができます。"all"の場合は、全ての PID のデータが表示されます。

- ⑤ 表示するデータのカテゴリID を指定します。十進数の数字、もしくは"all"を入力することができます。"all"の場合は、全てのカテゴリID のデータが表示されます。
- ⑥ 表示するデータのリージョンを指定します。それぞれのリージョンを ON,OFF することができます。
  - ・ JAPAN … 日本地域です。
  - ・ US … 北米地域です。
  - ・ EUROPE … 欧州地域です。
  - ・ KOREA … 韓国地域です。
- ⑦ データ表示ボタンです。このボタンを押すと、指定した条件のデータが ⑫ のリストに表示されます。
- ⑧ 前を表示ボタンです。オフセットが 0 より大きいときにデータを表示したときに有効になります。このボタンを押すと、現在のオフセットから 100 を引いた値のオフセットでデータを表示します。
- ⑨ 次を表示ボタンです。表示しているデータの後の順位のデータがあるときに有効になります。このボタンを押すと、現在のオフセットに 100 を足した値のオフセットでデータを表示します。
- ⑩ 表示するデータのオフセット指定テキストボックスです。ここで指定した順番から 100 個のデータを表示します。
- ⑪ 指定した条件のデータが何個あるかを表示します。データを表示すると更新されます。
- ⑫ データリストです。取得したデータをリストで表示します。

## 3.8 メニュー構成

### 3.8.1 File メニュー

- ・ Exit … アプリケーションを終了します。

### 3.8.2 Edit メニュー

- ・ Copy … データリストで選択したデータの内容を、クリップボードにコピーします。
- ・ Select All … データリストに表示されているデータを、全て選択します。

### 3.8.3 Operation メニュー

- ・ get CSV … CSVファイルを取得するGet CSVダイアログボックスを開きます。(3.9.1)
- ・ get UserData … ユーザー定義データを取得するGet UserDataダイアログボックスを開きます。(3.9.2)
- ・ Delete Entry … データを削除するための、Delete Entryダイアログボックスを開きます。(3.9.3)

### 3.8.4 Setting メニュー

- ・ use UTC … 有効にすると、時間を表示する際に、UTC（世界標準時）を使います。(3.5)
- ・ Proxy Setting … プロキシサーバを設定するProxy Server Settingsダイアログボックスが開きます。(3.9.4)
- ・ Public Server … 有効にすると、製品向けランキングサーバへ接続します。

### 3.8.5 Help メニュー

- Version … バージョン情報を表示します。

## 3.9 ダイアログボックス

### 3.9.1 Get CSV ダイアログボックス

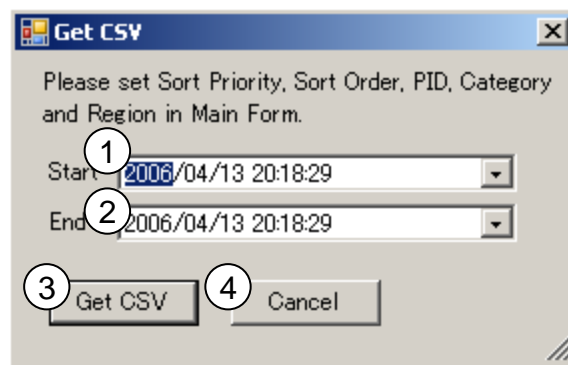


図 3-3 Get CSV ダイアログボックス

データを CSV ファイル形式で取得するためのダイアログボックスです。CSV ファイルを取得する条件のうち、ソート順、PID、カテゴリーID、リージョンはメインフォームの指定がそのまま使われます。

- ① 指定した日時からのデータを取得します。
- ② 指定した日時までのデータを取得します。
- ③ CSV ファイルの取得を開始します。
- ④ キャンセルします。ダイアログが閉じ、CSV ファイルは取得されません。

取得が終了するとファイル保存ダイアログボックスが開きますので、保存先を指定してください。

CSV ファイルは、一行が1エントリの以下のような形式で取得されます。

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

タブ区切りで、PID、カテゴリーID、最終更新時刻、スコア、base64 エンコードしたユーザー定義データの順番になります。この CSV ファイルで使用されている base64 エンコードは、‘+’ が ‘-’ に、‘/’ が ‘\_’ に置き換えられておりますので、ご注意ください。

CSV ファイルの取得には、一回で 5000 エントリまでという制限があります。制限を越えてしまった場合は警告が出ますので、時間を短くするなどして取得数を減らしてから再取得してください。

取得する数が多いと、完了するまでに1分以上時間がかかる場合があります。CSV の取得は非常に重い処理ですので、30分に1回程度の利用としてください。

### 3.9.2 Get UserData ダイアログボックス

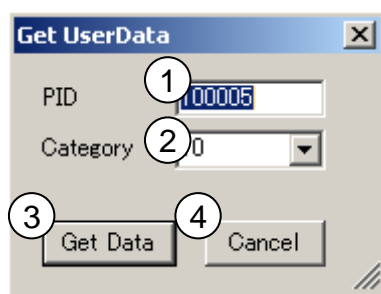


図 3-4 Get UserData ダイアログボックス

ユーザー定義データを取得するダイアログボックスです。データはバイナリ形式でファイルに保存されます。

- ① 取得するデータの PID を指定します。ダイアログボックスを開くときに、データリストでデータを選択していると、選択したデータの PID が入った状態でダイアログボックスが開きます。
- ② 取得するデータの 카테고리-ID を指定します。ダイアログボックスを開くときに、データリストでデータを選択していると、選択したデータの 카테고리-ID が入った状態でダイアログボックスが開きます。"all"は指定できません。
- ③ ユーザー定義データファイルの取得を開始します。
- ④ ユーザー定義データファイルの取得を取り消し、メインフォームへ戻ります。

取得が終了するとファイル保存ダイアログボックスが開きますので、保存先を指定してください。

### 3.9.3 Delete Entry ダイアログボックス

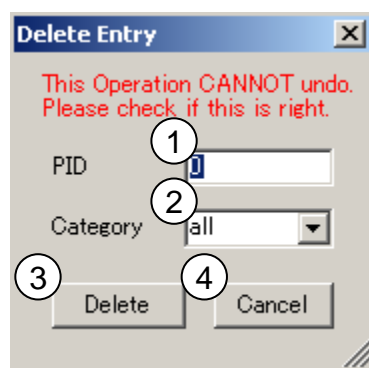


図 3-5 Delete Entry ダイアログボックス

データを削除するダイアログボックスです。削除は取り消すことができませんので、ご注意ください。

- ① 削除するデータの PID を指定します。ダイアログボックスを開くときに、データリストでデータを選択していると、選択したデータの PID が入った状態でダイアログボックスが開きます。開発向けサーバでは、"all"を選択することで全ての PID について処理を行うことができます。
- ② 削除するデータの 카테고리-ID を指定します。ダイアログボックスを開くときに、データリストでデータを選択していると、選択したデータの 카테고리-ID が入った状態でダイアログボックスが開きます。"all"を指定すると、全ての 카테고리-ID について削除します。
- ③ データの削除を開始します。
- ④ データの削除を取り消し、メインフォームへ戻ります。

データを削除すると、データリストの表示が自動的に更新されます。

### 3.9.4 Proxy Server Settings ダイアログボックス

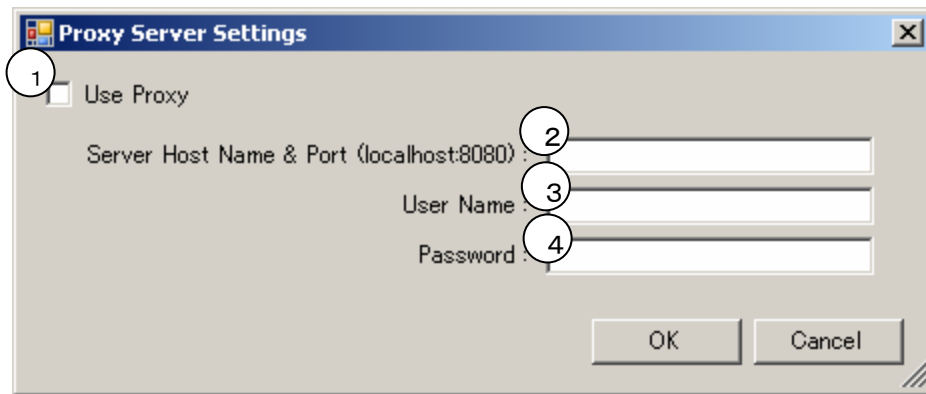


図 3-6 Proxy Server Settings ダイアログボックス

プロキシサーバを設定するダイアログボックスです。

- ① チェックすると指定したプロキシサーバを使用します。チェックしなかった場合は、インターネットエクスプローラーの設定が使われます。インターネットエクスプローラーの設定ではプロキシ認証を行うことが出来ないため、プロキシ認証が必要な場合にはここで設定する必要があります。
- ② プロキシサーバのホスト名とポート番号を指定します。
- ③ プロキシ認証を使用する場合のユーザー名を指定します。プロキシ認証を使用しない場合は空白にします。
- ④ プロキシ認証を使用する場合のパスワードを指定します。プロキシ認証を使用しない場合は空白にします。

※注意 ここで設定したユーザー名とパスワードは、管理ツールの設定ファイルに保存され、次回起動時にも有効になりますが、設定ファイルには暗号化が施されないため、セキュリティ的に問題があります。今後のバージョンアップで対策が行われるまではこの点を留意した上でご使用ください。

## 3.10 ショートカットキー

汎用ランキング管理ツールでは、以下のショートカットキーを使用することができます。

- ・ CTRL+C … データリストで選択したデータの内容を、クリップボードにコピーします。
- ・ CTRL+A … データリストに表示されているデータを、全て選択します。
- ・ CTRL+S … CSV ファイルを取得するための、Get CSV ダイアログボックスを開きます。
- ・ CTRL+U … ユーザー定義データを取得するための、Get UserData ダイアログボックスを開きます。
- ・ CTRL+D … データを削除するための、Delete Entry ダイアログボックスを開きます。

## 4 Web サービス開発

### 4.1 Web サービスについて

---

汎用ランキングライブラリは、収集されたデータをインターネット経由で取得するインターフェースを持っています。この機能を用いると、ランキングデータをゲームのホームページで表示する等のサービスを提供することができます。

### 4.2 使い方

---

#### 4.2.1 アクセス方法

---

汎用ランキングライブラリのデータを取得するには、以下のアドレスにアクセスします。HTTP のレスポンスとして、タブ区切りの CSV ファイルを取得することができます。

- ・ 製品向け

`http://gamestats2.gs.nintendowifi.net/[ゲーム名]/web/admin/getcsv.asp`

- ・ 開発向け

`http://sdkdev.gamespy.com/games/[ゲーム名]/web/admin/getcsv.asp`

※[ゲーム名]とはゲームタイトルにつき1つ割り当てられるユニークな文字列で、返却されたデザインステートメントに記載されています。

#### 4.2.2 セキュリティについて

---

汎用ランキングライブラリの Web サービスは、セキュリティのために IP アドレスによるアクセス制限を行っています。運用の際は、弊社担当窓口へアクセス元のグローバル IP アドレスを連絡してください。

なお、開発向けサーバではアクセス制限は行っていません。

#### 4.2.3 データの形式について

---

汎用ランキングライブラリの Web サービスで得られるデータはタブ区切りの CSV ファイルで、一行が一つのエントリを表しています。

要素の順番は、GS プロファイル ID、カテゴリーID、リージョン、最終更新時刻、スコア、base64 エンコードしたユーザー定義データとなります。この CSV ファイルで使用されている base64 エンコードは、`+` が `-` に、`/` が `_` に置き換えられておりますので、ご注意ください。

#### 4.2.4 サーバへの負荷について

---

Web サービスの利用は、ランキングサーバに大きな負荷をかけます。データの取得は 30 分に 1 回程度とし、取得パラメータの `since` を適切に利用して、同じデータを取得しないようにしてください。



### 4.2.5 取得パラメータについて

---

ランキングサーバへのリクエストにパラメータを与えることで、取得方法を指定することができます。  
リクエストパラメータを与えるには、URL の後に以下のような形式の文字列を付加します。

`http://.../getcsv.asp?[パラメータ名 1]=[値 1]&[パラメータ名 2]=[値 2]&...`

指定できるパラメータと値は、以下の通りです。パラメータは全て指定する必要はなく、指定しなかった場合はデフォルト値が使用されます。

**sort** ... データのソートの仕方を指定します。デフォルトは 0 です。

0: 取得するデータをスコアの昇順でソートします。

1: 取得するデータをスコアの降順でソートします。

**region** ... 取得するデータのリージョンを指定します。以下の数値の OR になります。

デフォルトは 255（全て）です。

1: 日本

2: 北米

4: 欧州

8: 韓国

**pid** ... 取得するデータの PID を数値で指定します。指定しないと、全ての PID についてデータを取得します。

**category** ... 取得するデータの 카테고리ID を数値で指定します。指定しないと、全ての 카테고리ID についてデータを取得します。

**limit** ... 取得するデータの最大数を数値(1～5000)で指定します。  
デフォルトは 100 となります。

**since** ... 指定された日時以降に更新されたデータのみを取得します。

デフォルトでは、日時での絞り込みは行いません。

フォーマットは [年]-[月]-[日]-[時]-[分]-[秒] となります。

UTC（世界標準時）で指定してください。

例えば、2006 年 4 月 1 日 13 時以降に更新された、カテゴリ ID が 10 のデータを、最大 1000 個取得したい場合は、次のようにパラメータを指定します。

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

記載されている会社名、製品名等は、各社の登録商標または商標です。

© 2006-2007 Nintendo

任天堂株式会社の許諾を得ることなく、本書に記載されている内容の一部あるいは全部を無断で複製・複写・転写・頒布・貸与することを禁じます。