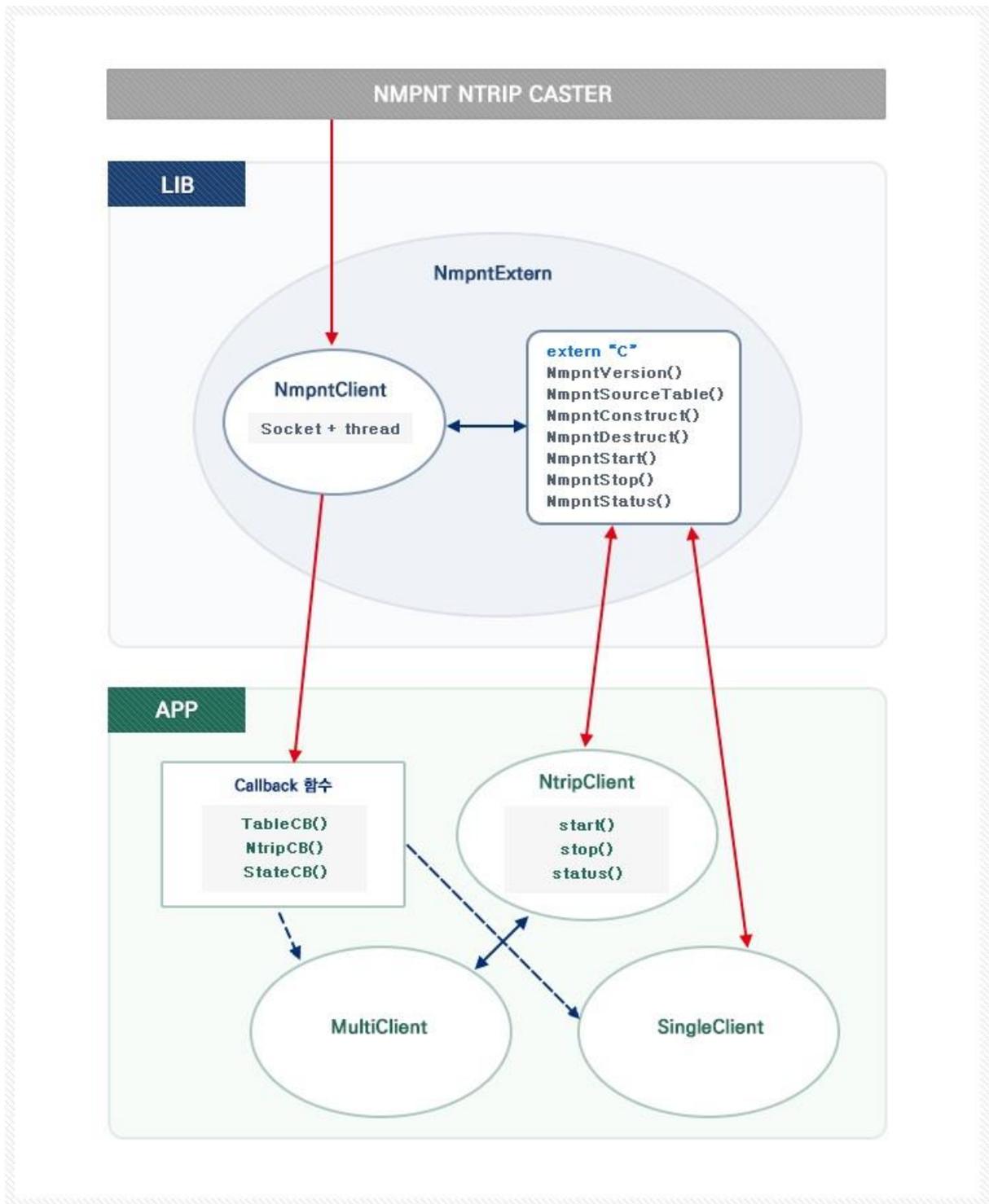


# NmpntClient 라이브러리 사용설명서

NmpntClient 라이브러리는 NTRIP 클라이언트를 위한 API를 제공하는 라이브러리로 윈도우와 리눅스를 지원합니다. 라이브러리는 NTRIP 프로토콜과 스레드를 NmpntClient 클래스로 구현하고, NmpntExtern 파일에 사용할 수 있는 API를 나열하였습니다.

다음 <그림>은 NTRIP 클라이언트 라이브러리와 응용프로그램의 관계를 보여주며, 라이브러리 헤더파일은 NmpntExtern.h와 윈도우의 NmpntClient.lib, NmpntClient.dll과 리눅스의 libNmpntClient64.so, libNmpntClient32.so가 제공되며, 사용예제는 라이브러리 API를 직접 사용하는 SingleClient와 여러 클라이언트를 동시에 처리할 수 있게 클래스로 구현한 MultiClient를 제공합니다.

라이브러리 API 사용시 주의해야할 사항은 thread-safe 구조로, **라이브러리로부터 호출된 콜백함수에서 라이브러리 API를 호출하게 되면 교착상태(deadlock)에 빠질 위험성**을 내포하고 있습니다. 윈도우는 MSVC 2015 환경에서 빌드되었으며, 응용프로그램은 시작/종료시 WSASStartup()과 WSACleanup() 함수를 이용하여 응용프로그램 내에서 소켓을 사용할 수 있는 환경을 만들어야 합니다. 리눅스는 GNU(g++) 4.4.7 에서 빌드되었으며 개발환경에 따라 요구되는 패키지 설치가 필요할 수 있습니다. 응용프로그램은 buildNtrip 스크립트를 이용하여 매개변수 64 또는 32를 주면 64비트/32비트 응용프로그램이 생성됩니다.



<그림> NmpntClient 라이브러리와 응용프로그램 관계

# NmpntVersion

---

## 구문

```
LibVersion NmpntVersion(void);
```

## 반환값

NmpntClient 라이브러리 버전값(LibVersion 형식)

## 설명

현재 사용중인 NmpntClient 라이브러리 버전 정보를 조회하기 위한 함수입니다.

## 예제

```
typedef struct {
    unsigned int    major : 8; // 1 byte
    unsigned int    minor : 8; // 1 byte
    unsigned int    patch : 16; // 2 byte
} LibVersion; // 4 bytes

int main()
{
    ...
    // version 체크
    LibVersion    ver = NmpntVersion();
    printf("library version : %04d.%02d.%02d\n", ver.major, ver.minor, ver.patch);
    ...
}
```

# NmpntSourceTable

---

## 구문

```
int NmpntSourceTable(
    TableCB *table
);
```

## 매개변수

*table*

수신된 소스 테이블의 마운트포인트를 전달받기 위한 콜백함수 포인터

## 반환값

조회된 마운트포인트의 개수

## 설명

캐스터에서 제공하는 마운트포인트를 조회하기 위한 함수입니다.

## 예제

```
// Callback Functions
typedef void(*TableCB)(char *mount); // Source Table

int main()
{
    ...
    // source table 체크
    printf("mountpoint count : %d\n", NmpntSourceTable(&table));
    ...
}
```

# NmpntConstruct

---

## 구문

```
NmpntClient *NmpntConstruct(void)
```

## 반환값

생성된 NmpntClient의 클래스 객체 포인터

## 설명

윈도우 프로그램에서만 사용되는 함수로 한 개의 마운트포인트에 대해서 최소 한 개 이상을 할당하여 사용하여야 합니다.

## 요구사항

생성된 NmpntClient 객체는 사용 후에 **반드시 소멸함수(NmpntDestruct)로 정리**하여야 합니다.

## 예제

```
NtripClient::NtripClient()  
{  
    _client = NmpntConstruct();  
    ...  
}
```

# NmpntDestruct

---

## 구문

```
void NmpntDestruct(  
    NmpntClient *nmpnt  
)
```

## 매개변수

*nmpnt*

소멸하기 위한 NmpntClient 클래스 객체 포인터

## 설명

할당된 NmpntClient 객체를 소멸하기 위한 함수입니다.

## 예제

```
NtripClient::~NtripClient()  
{  
    stop();  
    NmpntDestruct(_client);  
}
```

# NmpntStart

---

## 구문

```
void NmpntStart(  
    NmpntClient *nmpnt,  
    void *owner,  
    char *mount,  
    int purpose,  
    NtripCB ntrip,  
    StateCB state  
);
```

## 매개변수

*nmpnt*

라이브러리 영역의 할당된 객체를 참조하기 위한 `NmpntClient` 객체 포인터

*owner*

응용프로그램 영역에서 구분되기 위한 객체 포인터

*mount*

마운트포인트명의 문자 포인터

*purpose*

마운트포인트의 사용목적(정의참조)

*ntrip*

수신된 NTRIP 데이터를 전달받기 위한 콜백함수 포인터

*state*

`NmpntClient` 객체의 상태를 전달받기 위한 콜백함수 포인터

## 설명

지정된 마운트포인트의 NTRIP 데이터를 수신하기 위한 스레드를 시작하기 위한 함수입니다.

## 예제

```
// Callback Functions  
typedef void(*NtripCB)(void *owner, char *data, int size); // Receive Data  
typedef void(*StateCB)(void *owner, int state);           // Status Event  
  
// Client Use Purpose  
#define NTRIP_PURPOSE_ETC      0 // 기타  
#define NTRIP_PURPOSE_MES      1 // 측량  
#define NTRIP_PURPOSE_GIS      2 // FGIS/GIS  
#define NTRIP_PURPOSE_RND      3 // 연구개발  
#define NTRIP_PURPOSE_ENV      4 // 환경조사  
#define NTRIP_PURPOSE_CON      5 // 건설  
#define NTRIP_PURPOSE_TRF      6 // 교통  
#define NTRIP_PURPOSE_ANF      7 // 농/수산업
```

```

#define NTRIP_PURPOSE_COM      8   // 통신
#define NTRIP_PURPOSE_EXP      9   // 원격탐사
#define NTRIP_PURPOSE_TMP     10  // 기상
#define NTRIP_PURPOSE_EDU     11  // 교육
#define NTRIP_PURPOSE_AIR     12  // 항공

void NtripClient::start(char *mountpoint, int purpose, unsigned int result)
{
    char          fname[256];

    strncpy(_mountpoint, mountpoint, 31);
    _result = result;

    stop();
    if (_result & NTRIP_FILESAVE)
    {
        sprintf(fname, "./%s.rtc", _mountpoint);
        _file = fopen(fname, "wb");
    }

    if (_result & NTRIP_DECODE)
    {
        //...
    }

    if (_result & NTRIP_NETWORK)
    {
        //...
    }

    if (_result & NTRIP_SERIAL)
    {
        //...
    }

    NmpntStart(_client, this, _mountpoint, purpose, &ntrip, &state);
}

```

# NmpntStop

---

## 구문

```
void NmpntStop(  
    NmpntClient *nmpnt  
);
```

## 매개변수

*nmpnt*

라이브러리 영역의 할당된 객체를 참조하기 위한 NmpntClient 객체 포인터

## 설명

지정된 마운트포인트의 스레드를 정지하기 위한 함수입니다.

## 예제

```
void NtripClient::stop()  
{  
    NmpntStop(_client);  
  
    if (_file)  
        fclose(_file);  
    _file = NULL;  
}
```

# NmpntState

---

## 구문

```
int NmpntStatus(  
    NmpntClient *nmpnt  
);
```

## 매개변수

*nmpnt*

라이브러리 영역의 할당된 객체를 참조하기 위한 NmpntClient 객체 포인터

## 설명

스레드로 동작하는 NmpntClient 객체의 상태를 조회하기 위한 함수입니다.

## 예제

```
// Client Status Code  
#define NTRIP_ONAIR          1 // 서비스 정상  
#define NTRIP_OFFAIR        0 // 서비스 정지  
#define NTRIP_SOCKET_ERROR -1 // 소켓 오류  
#define NTRIP_CONNECT_ERROR -2 // 연결 오류  
#define NTRIP_TIMEOUT_ERROR -3 // 수신 불량  
#define NTRIP_CONNECT_CLOSE -4 // 연결 끊김  
#define NTRIP_MOUNT_ERROR   -5 // 마운트명 오류  
#define NTRIP_USER_ERROR    -6 // 사용자정보 오류  
#define NTRIP_VERSION_ERROR -7 // 버전 오류  
#define NTRIP_CASTER_ERROR  -8 // 캐스터 오류  
  
void NtripClient::status()  
{  
    printf("status : %s / %d\n", _mountpoint, NmpntStatus(_client));  
}
```

# TableCB

---

## 구문

```
void *TableCB(  
    char *mount  
);
```

## 매개변수

*mount*

조회된 마운트포인트의 문자 포인터

## 설명

캐스터 소스태이블을 수신하여 마운트포인트를 콜백함수(TableCB)로 전달합니다.

## 예제

```
static void table(char *mount)  
{  
    //...  
    printf("%s\n", mount);  
}
```

# NtripCB

---

## 구문

```
void *NtripCB(
    void *owner,
    char *data,
    int size
);
```

## 매개변수

*owner*

콜백함수를 호출한 객체를 소유하고 있는 응용프로그램 영역의 객체

*data*

지정된 마운트포인트로부터 수신된 NTRIP 데이터 포인터

*size*

수신된 데이터의 크기

## 설명

지정된 마운트포인트의 NTRIP 데이터가 수신되면 콜백함수(NtripCB)로 전달합니다.

## 예제

```
static void ntrip(void *self, char *data, int size)
{
    NtripClient    *owner = (NtripClient *)self;

    printf("%s : ntrip : %d\n", owner->_mountpoint, size);
    if (owner->_result & NTRIP_FILESAVE)
    {
        //...
        if (owner->_file)
            fwrite(data, 1, size, owner->_file);
    }

    if (owner->_result & NTRIP_DECODE)
    {
        if (owner->_mountpoint[9] == '2') // RTCM 2.x
        {
            //...
        }
        else //if (owner->_mountpoint[9] == '3') // RTCM 3.x
        {
            //...
        }
    }
    ...
}
```

# StateCB

---

## 구문

```
void *StateCB(
    void *owner,
    int state
);
```

## 매개변수

*owner*

콜백함수를 호출한 객체를 소유하고 있는 응용프로그램 영역의 객체

*state*

NmpntClient 객체의 스레드 상태값(정의참조)

## 설명

스레드로 동작하는 NmpntClient 객체의 상태가 변경되면 상태정보를 콜백함수(StateCB)로 전달합니다.

## 예제

```
// Client Status Code
#define NTRIP_ONAIR          1 // 서비스 정상
#define NTRIP_OFFAIR        0 // 서비스 정지
#define NTRIP_SOCKET_ERROR -1 // 소켓 오류
#define NTRIP_CONNECT_ERROR -2 // 연결 오류
#define NTRIP_TIMEOUT_ERROR -3 // 수신 불량
#define NTRIP_CONNECT_CLOSE -4 // 연결 끊김
#define NTRIP_MOUNT_ERROR   -5 // 마운트명 오류
#define NTRIP_USER_ERROR    -6 // 사용자정보 오류
#define NTRIP_VERSION_ERROR -7 // 버전 오류
#define NTRIP_CASTER_ERROR  -8 // 캐스터 오류

static void state(void* self, int status)
{
    printf("%s : ", ((NtripClient *)self)->_mountpoint);
    switch (status)
    {
        case NTRIP_ONAIR:
            printf("서비스를 시작하였습니다.\n");
            break;
        case NTRIP_OFFAIR:
            printf("서비스를 정지하였습니다.\n");
            break;
        ...
    }
}
```