

# BaroPAM 가이드(ASP)

## 목차

목차.....	0
1. BaroPAM 연동 API.....	1
1.1 연동 API 구성.....	1
1.2 연동 API 함수.....	2
1.3 인증키 검증 부분.....	2
2. BaroPAM 적용.....	9
2.1 BaroPAM 적용 프로세스.....	9
2.2 BaroPAM 적용 화면.....	9
2.3 본인확인 적용 프로세스.....	10
2.4 본인확인 적용 화면.....	11
2.5 BaroPAM 앱 설치 및 정보 설정.....	12
3. NTP(Network Time Protocol) 설정.....	13
3.1 Windows 환경.....	13
4. About BaroPAM.....	22

# 1. BaroPAM 연동 API

## 1.1 연동 API 구성

BaroPAM 관련 COM DLL은 ASP 환경에서 **일회용 인증키**를 검증하는데 사용된다.

API구분	설명	비고
barokey.h barokey_asp.dll	BaroPAM 관련 Header 파일과 ASP용 COM dll	
libcrypto-1_1-x64.dll libssl-1_1-x64.dll	Open SSL 관련 dll	

참고) BaroPAM 관련 dll을 사용하기 위해서는 반드시 "C:\Windows\System32" 디렉토리에 위치해야 한다.

**일회용 인증키**인 BaroPAM에 대한 header 파일인 barokey.h은 다음과 같다.

```
#ifndef _BAROKEY_API_H_
#define _BAROKEY_API_H_

#ifdef BAROPAMCORE_EXPORTS
#define BAROPAMCORE_API __declspec(dllexport)
#else
#define BAROPAMCORE_API __declspec(dllimport)
#endif

#ifdef __cplusplus
extern "C" {
#endif

BAROPAMCORE_API BOOL BARO_ENCRYPT(const char* data, char* enc_result, unsigned long buf_len);
BAROPAMCORE_API BOOL BARO_DECRYPT(const char* data, char* dec_result, unsigned long buf_len);
BAROPAMCORE_API BOOL BARO_GENERATEKEY(const char* login_id, const char* phone_no, const char*
cycle_time, char* ota_key, unsigned long buf_len);
BAROPAMCORE_API BOOL BARO_VERIFYKEY(const char* login_id, const char* phone_no, const char*
cycle_time, char* ota_key);

BAROPAMCORE_API char* BARO_ENCRYPTA(const char* data);
BAROPAMCORE_API char* BARO_DECRYPTA(const char* data);
BAROPAMCORE_API char* BARO_GENERATEKEYA(const char* login_id, const char* phone_no, const char*
cycle_time);
BAROPAMCORE_API bool BARO_VERIFYKEYA(const char* login_id, const char* phone_no, const char*
cycle_time, char* ota_key);

BAROPAMCORE_API wchar_t* BARO_ENCRYPTSW(const wchar_t* data);
BAROPAMCORE_API wchar_t* BARO_DECRYPTSW(const wchar_t* data);
BAROPAMCORE_API wchar_t* BARO_GENERATEKEYW(const wchar_t* login_id, const wchar_t* phone_no,
const wchar_t* cycle_time);
BAROPAMCORE_API bool BARO_VERIFYKEYW(const wchar_t* login_id, const wchar_t* phone_no, const
wchar_t* cycle_time, wchar_t* totp);
```

```
#ifdef __cplusplus
}
#endif

#endif // _BAROKEY_API_H_
```

## 1.2 연동 API 함수

### 1) BARO\_VERIFYKEYA / BARO\_VERIFYKEYW 함수

#### - NAME

BARO\_VERIFYKEYA / BARO\_VERIFYW (UNICODE 용)

#### - SYNOPSIS

```
bool BARO_VERIFYKEYA(const char* login_id, const char* phone_no, const char* cycle_time, char*
ota_key)
```

```
bool BARO_VERIFYKEYW(const wchar_t* login_id, const wchar_t * phone_no, const wchar_t *
cycle_time, wchar_t * ota_key)
```

#### - DESCRIPTION

입력한 **일회용 인증키**가 맞는지 검증하는 함수.

login\_id: 로그인-ID 항목에 입력한 ID를 설정.

phone\_no: 사용자별 스마트 폰 번호를 숫자만 설정.

cycle\_time: 사용자별로 지정한 **일회용 인증키**의 생성 주기(3-60초)를 설정.

ota\_key: BaroPAM 앱에서 생성하여 입력한 **일회용 인증키**를 설정.

만약, 사용자별로 스마트 폰 번호 및 개인별로 지정한 **일회용 인증키**의 생성 주기가 **일회용 인증키**의 생성기와 다른 경우 **일회용 인증키**가 달라서 검증에 실패할 수 있다. 반드시 정보를 일치시켜야 한다.

#### - RETURN VALUES

성공 시에는 **true**을 반환하며, 실패 시는 **false**을 반환.

## 1.3 인증키 검증 부분

Sample program) ASP에서 DLL을 직접 호출하는 경우

```
<%
dim login_id, passwd, ota_key

login_id = Request.Form("login_id")
passwd = Request.Form("passwd")
ota_key = Request.Form("ota_key")

// phone_no, cycle_time은 login_id별로 사용자DB에서 가져옴.
Set phone_no = "01027714076"
Set cycle_time = "30"
```

```

// 객체 생성
Set obj = Server.CreateObject("barokey.barokey")

// 개체 생성(성공)
If isObject(obj) Then
    // OTA key 검증 함수 호출
    bota_key = obj.BARO_VERIFYKEY(login_id, phone_no, cycle_time, ota_key)

    // OTA key 검증(성공)
    If bota_key = true Then
        Response.Write("Auth key success.")

    // OTA key 검증(실패)
    else
        Response.Write("Auth key faild.")
    end if

// 개체 생성(실패)
else
    Response.Write("Object creation failed.")
end if

Set obj = Nothing
%>

```

예) cmd 창에서 다음 명령어를 이용해 dll 파일을 서버에 등록.

```
regsvr32 barokey_asp.dll
```

예) cmd 창에서 다음 명령어를 이용해 dll 파일을 서버에서 해제

```
regsvr32 /u barokey_asp.dll
```

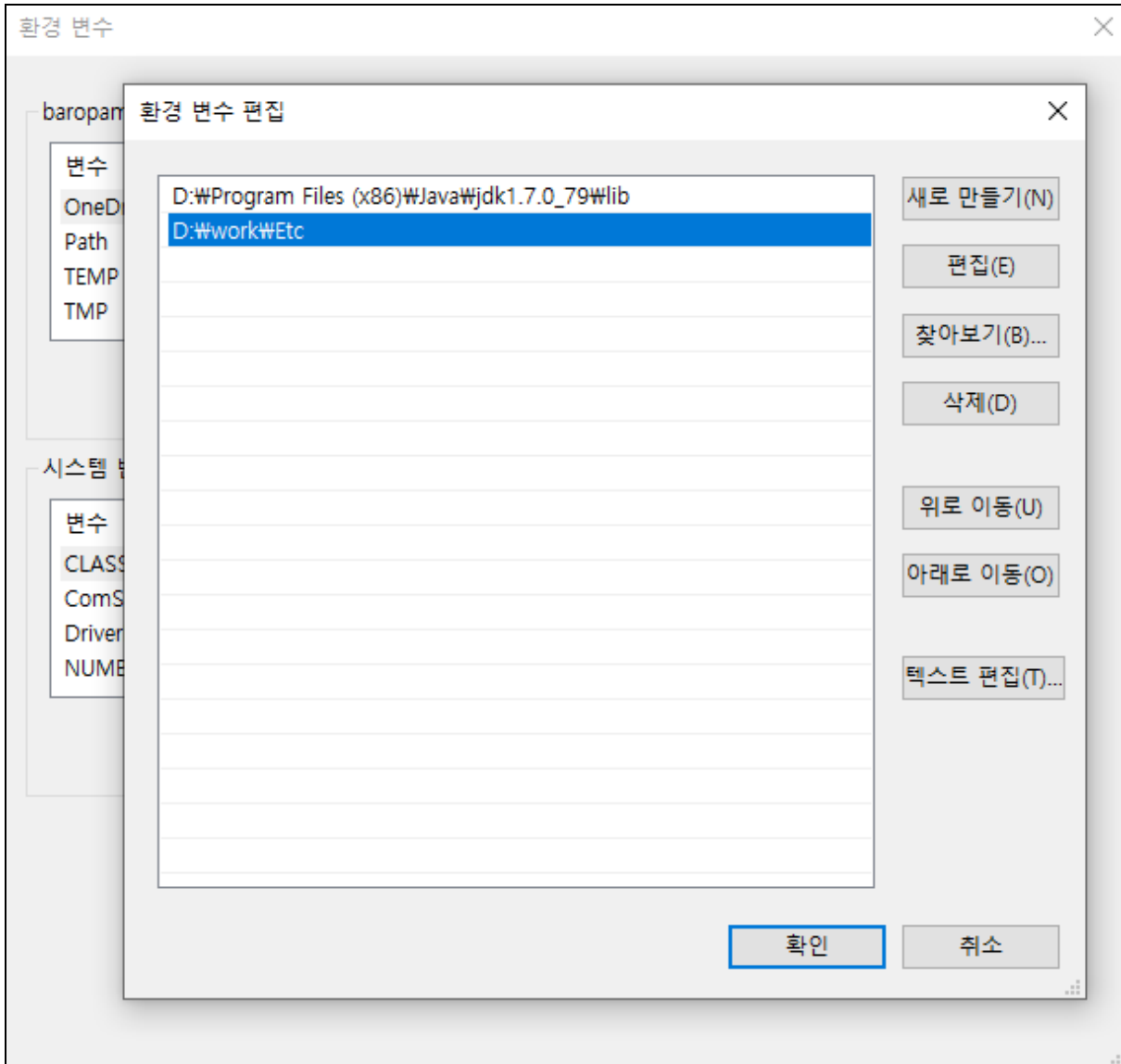
예) dll 파일을 서버에 등록/해제한 다음 반드시 cmd 창에서 다음 명령어를 이용해서 iis 관리 서비스 재 기동 해야 함.

```
net stop iisadmin
net start iisadmin
net start w3svc
```

Sample program) Java Class인 barokey.class를 호출하는 경우

작업순서)

1) 환경변수(CLASSPATH) 설정(D:\work\etc 디렉토리에 barokey.class 존재)



2) SDKJava40.exe 다운로드 및 설치

Java용 Microsoft SDK(SDKJava)는 Microsoft Internet Explorer 환경용 개발자 키트임. JDK 1.1 및 Microsoft VM(Microsoft Win32 가상 머신)을 기반으로 Java 프로그램 및 애플릿을 개발하는 데 도움이 되는 도구, 정보, 샘플이 제공됨.

다운로드: [https://pt.osdn.net/projects/sfnet\\_webuniproj/downloads/ms-jsdk4/sdk/SDKJava40.exe/](https://pt.osdn.net/projects/sfnet_webuniproj/downloads/ms-jsdk4/sdk/SDKJava40.exe/)

3) 레지스터 등록

```
C:\work\Etc> C:\Program Files (x86)\Microsoft SDK for Java 4.0\Bin\javareg /register /class:barokey.barkey /progid:barokey.barkey
```

정상적으로 레지스터리에 등록되면 다음과 같은 메시지가 나타난다.



4) asp 프로그램에서 verifyKEY() 메소드 호출

```
<%
dim login_id, passwd, ota_key

login_id = Request.Form("login_id")
passwd = Request.Form("passwd")
ota_key = Request.Form("ota_key")

// phone_no, cycle_time은 login_id별로 사용자DB에서 가져옴.
Set phone_no = "01027714076"
Set cycle_time = "30"

// 객체 생성
Set obj = Server.GetObject(java:"barokey")

// 개체 생성(성공)
If isObject(obj) Then
    // OTA key 검증 함수 호출
    bota_key = obj.verifyKEY(client, phone_no, cycle_time, ota)

    // OTA key 검증(성공)
    If bota_key = true Then
        Response.Write("Auth key success.")

    // OTA key 검증(실패)
    else
        Response.Write("Auth key faild.")
    end if

// 개체 생성(실패)
else
    Response.Write("Object creation failed.")
end if

Set obj = Nothing
%>
```

Sample program) Tomcat에 검증 모듈을 구성하는 경우

일회용 인증키 검증 요청)

```

<%
dim xmlHttp, postdat, login_id, phone_no, cycle_time, ota_key

login_id = Request.Form("login_id" )
phone_no = Request.Form("phone_no" )
cycle_time = Request.Form("cycle_time")
ota_key = Request.Form("ota_key" )

postdata = "login_id=" & login_id
postdata = postdata & "&phone_no="
postdata = postdata & phone_no
postdata = postdata & "&cycle_time="
postdata = postdata & cycle_time
postdata = postdata & "&ota_key="
postdata = postdata & ota_key

Set xmlHttp = CreateObject("Msxml2.XMLHTTP")

'개체 생성()
If IsObject(obj) Then
xmlHttp.Open "POST", "http://nuriapp.com/baropam/web/result_ota.jsp", False
xmlHttp.setRequestHeader "Content-Type", "application/x-www-form-urlencoded; charset=euc-kr"
xmlHttp.setRequestHeader "Content-Type", "application/x-www-form-urlencoded; charset=utf-8"
xmlHttp.Send postdata

'OTA key 검증(성공)
If StrComp(Mid(Cstr(xmlHttp.responseText), 17, 2), "00") = 0 Then
If StrComp(Replace(Cstr(xmlHttp.responseText), vbCrLf, ""), "00") = 0 Then
Response.Write("Authentication failed.")

'OTA key 검증(실패)
Else
Response.Write("Authentication failed.")
End if

'개체 생성(실패)
else
Response.Write("Object creation failed.")
end if

Set xmlHttp = nothing
%>

```

일회용 인증키 검증 결과)

BaroPAM에서 사용하는 인증 코드인 **일회용 인증키**는 Java를 기반으로 작성되었기 때문에 반드시 최신 **JDK 6.x 이상**이 설치되어 있어야 한다. 만약, 설치되어 있지 않으면 최신 JDK를 설치해야 한다.

**일회용 인증키**를 검증하는 API는 "**barokey.jar**"로 제공되며, WAS(Web application Server)의 lib 디렉토리 "**barokey.jar**"를 위치시키거나 classpath에 "**barokey.jar**"가 존재하는 디렉토리를 포함해서 설정해 주면 된다.

```
<@ page contentType="text/html; charset=UTF-8" language="java" pageEncoding="UTF-8" %>
```

```

<%@ page import="org.apache.log4j.*"%>
<%@ page import="java.text.*"%>
<%@ page import="java.util.*"%>
<%@ page import="com.barokey.*"%>
<% request.setCharacterEncoding("utf-8"); %>
<% response.setContentType("text/html; charset=utf-8"); %>
<%!
private Logger logger = Logger.getLogger("JspLogger");
%>
<%
/*-----*/
/* 변수선언 및 초기화. */
/*-----*/
int ii = 0, jj = 0, kk = 0, ll = 0; // Index

String result = ""; // Result
boolean bota_key = false; // 인증키 검증
/*-----*/
/* Request에서 데이터를 얻어옴(로그인 정보). */
/*-----*/
String login_id = request.getParameter("login_id");
String phone_no = request.getParameter("phone_no");
String cycle_time = request.getParameter("cycle_time");
String ota_key = request.getParameter("ota_key");

logger.info("(result_ota.jsp)Starting.....");
String param = request.getServerName() + request.getRequestURI()
+ "?remote_addr=" + request.getRemoteAddr()
+ "&login_id=" + login_id
+ "&phone_no=" + phone_no
+ "&cycle_time=" + cycle_time
+ "&ota_key=" + ota_key
;
logger.info(param);
/*-----*/
/* Begin. */
/*-----*/
try {
/*-----*/
/* 인증키 검증. */
/*-----*/
bota_key = barokey.verifyKEY(login_id, phone_no, cycle_time, ota_key);
/*-----*/
/* 인증키 검증(성공). */
/*-----*/
if (bota_key == true) {
result = "00";
/*-----*/
/* 인증키 검증(실패). */
/*-----*/
} else {
result = "99";
}
}
}

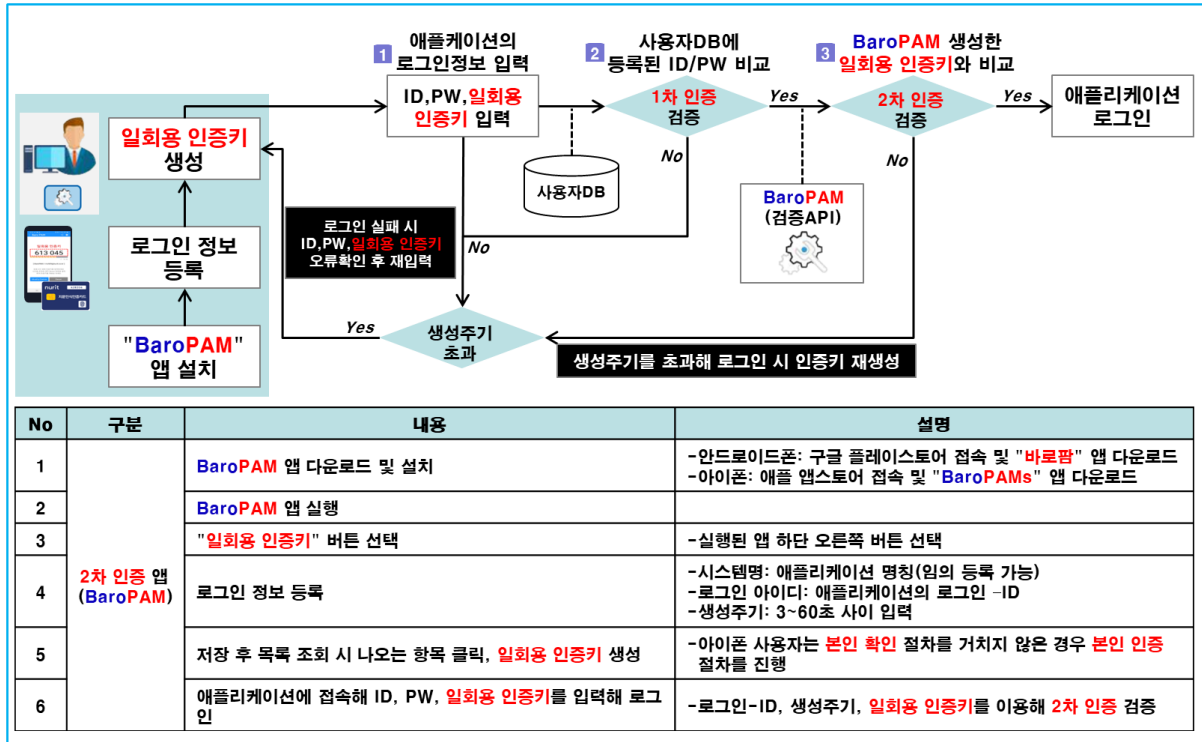
```



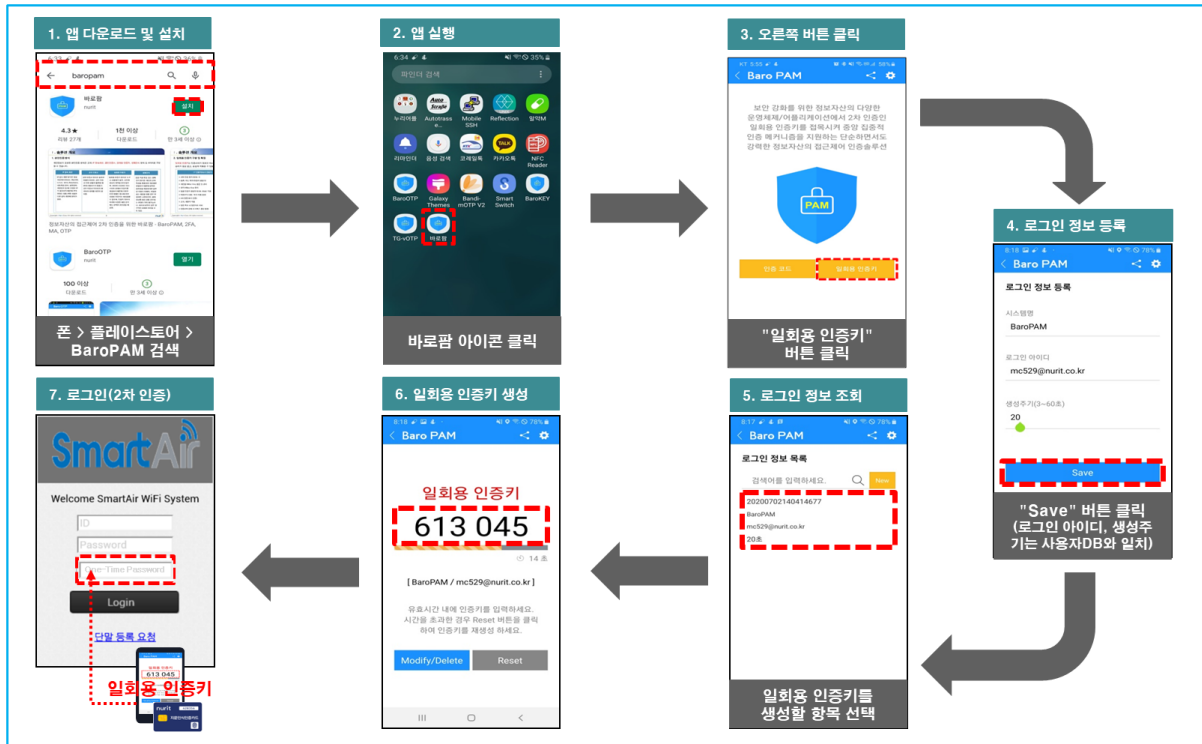
```
    out.println(result);
    /*-----*/
    /* 예외사항 처리(Exception).                */
    /*-----*/
} catch(Exception e) {
    logger.info("Exception = [" + e + "]);
    e.printStackTrace();
    /*-----*/
    /* Finally.                                */
    /*-----*/
} finally {
    logger.info("(result_ota.jsp)Ending.....");
}
%>
```

## 2. BaroPAM 적용

### 2.1 BaroPAM 적용 프로세스



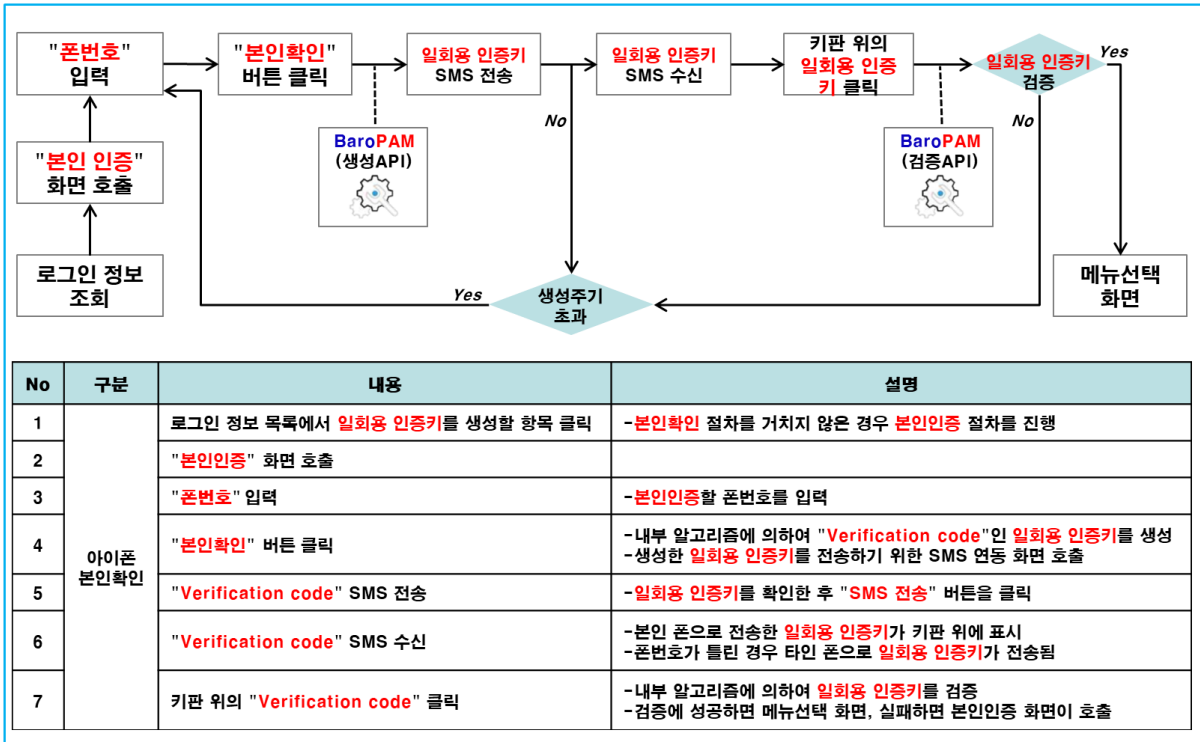
### 2.2 BaroPAM 적용 화면



### 2.3 본인확인 적용 프로세스

아이폰 (iPhone)의 기기정보를 얻지 못해서 **2차 인증키(일회용 인증키)**를 생성하기 위해서 로그인 정보 항목을 선택했을 때 "**일회용 인증키**" 생성 화면으로 이동하지 않은 경우가 발생할 수 있다.

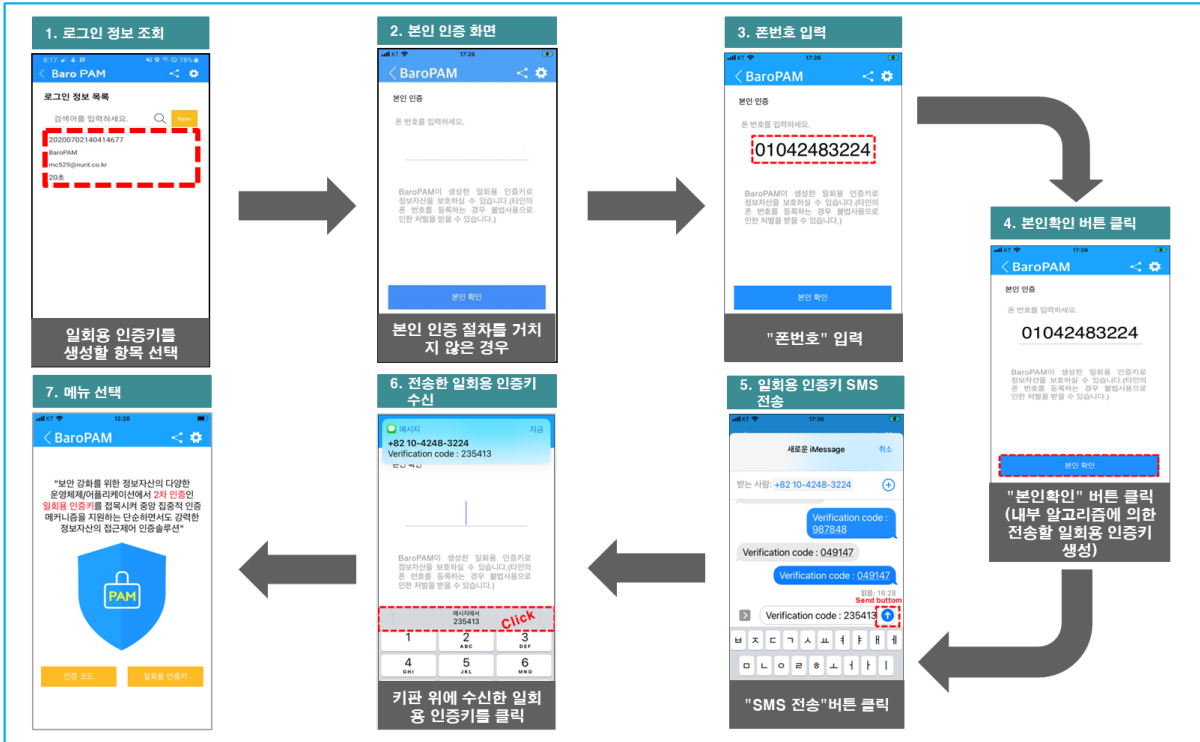
또한, 타인의 폰번호를 부정으로 사용하지 못하도록 하기 위해서 별도의 본인확인 기능을 적용할 필요가 있는데, "**BaroPAM**" 앱에서는 자체 알고리즘을 적용하여 자체적으로 본인확인 절차를 실행하고 있다.



## 2.4 본인확인 적용 화면

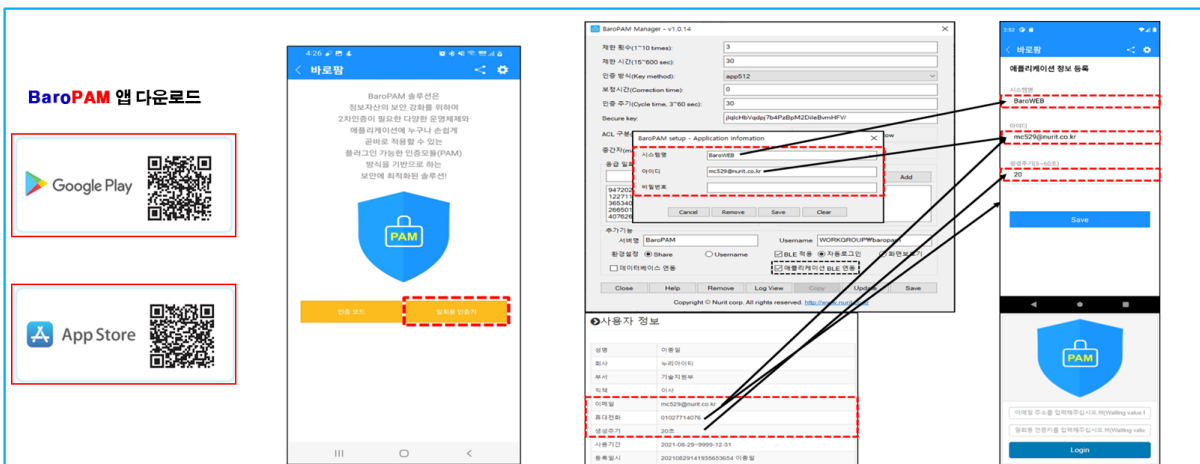
아이폰(iPhone)의 기기정보를 얻지 못해서 **2차 인증키(일회용 인증키)**를 생성하기 위해서 로그인 정보 항목을 선택했을 때 "**일회용 인증키**" 생성 화면으로 이동하지 않은 경우가 발생할 수 있다.

또한, 타인의 폰번호를 부정으로 사용하지 못하도록 하기 위해서 별도의 본인확인 기능을 적용할 필요가 있는데, "**BaroPAM**" 앱에서는 자체 알고리즘을 적용하여 자체적으로 본인확인 절차를 실행하고 있다.



## 2.5 BaroPAM 앱 설치 및 정보 설정

정보자산에 로그인 시 Verification code에 입력할 일회용 인증키의 생성기인 BaroPAM 앱의 다운로드 (<https://play.google.com/store/apps/details?id=com.baro.pam>)는 구글의 "Play 스토어"나 Apple의 "App 스토어"에서 가능하며, 설치의 일반 앱의 설치와 동일하다.



BaroPAM 앱은 Android 6.0 (Marshmallow) API 23, iOS 13.0 이상에서 사용 가능하며, 가로보기 모드를 지원하지 않는다.

BaroPAM 앱을 설치한 후 BaroPAM 앱을 실행하여 메뉴 선택화면에서 "일회용 인증키" 버튼을 클릭하여 애플리케이션의 사용자 정보에 설정한 "인증 주기, 아이디, 시스템명"을 BaroPAM 앱의 "애플리케이션 정보 등록" 화면에서 동일하게 입력해야 한다.

현상 : 안드로이드폰 또는 아이폰의 날짜와 시간이 현재 시간과 차이가 발생하여 "일회용 인증키"가 맞지 않은 경우

원인 : 안드로이드폰 또는 아이폰의 날짜와 시간이 네트워크에서 제공하는 시간을 사용하지 않아서 발생.

조치 : 안드로이드폰인 경우는 폰의 "설정" -> "일반" -> "날짜 및 시간" -> "날짜 및 자동 설정" 과 "시간대 자동 설정" -> "허용"  
아이폰인 경우는 폰의 "설정" -> "날짜 및 시간" -> "자동으로 설정" -> "허용"

### 3. NTP(Network Time Protocol) 설정

최근에는 서버/네트워크 장비에 대한 시간 동기화(타임서버 시간 동기화)하는 방법으로 NTP(Network Time Protocol)을 이용하여 관리자 계정에서 시스템의 시각을 현재 시각으로 설정할 수 있다.

#### 3.1 Windows 환경

##### 1. NTP란?

Network Time Protocol(네트워크 시간 프로토콜)의 약자로 네트워크 환경으로 구성된 장비(서버, PC, 통신장비, 방화벽 장비 등)의 시스템 시간을 동기화 하기 위한 규약이다.

윈도우에는

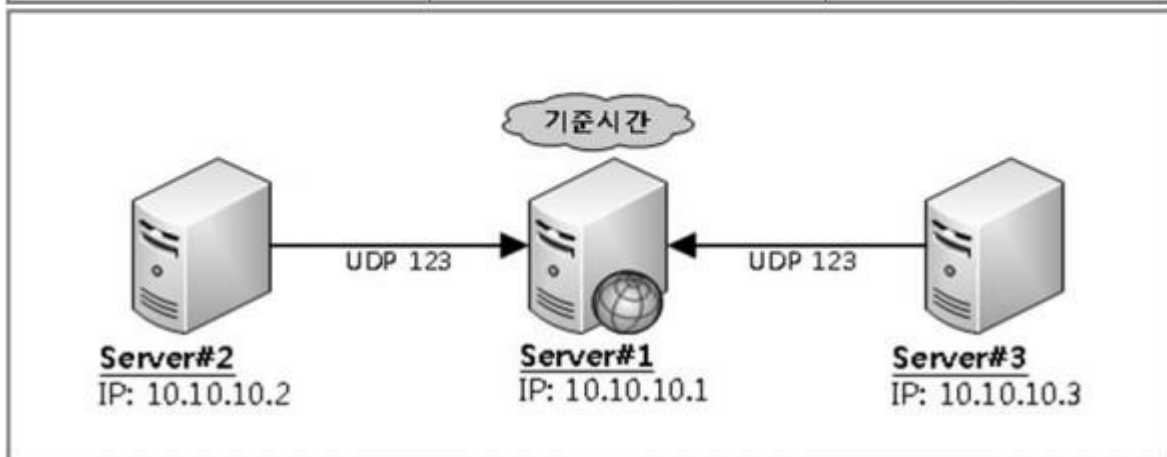
- 모든 윈도우서버는 NTP 서버가 될 수 있다.
- 모든 서버의 방화벽에서 UDP 123 번이 열려있어야 한다.
- NTP 서버와의 시간차이가 많이 날 경우(기본값: 15 시간) 동기화 되지 않는다.  
실제 시간과 근접한 시간으로 변경 후 동기화 할 수 있도록 한다.
- 데이터 일치를 위하여 동기화(Sync)요청을 하더라도 즉시 반영되는 것이 아니라, 조금씩 맞춰 간다.  
(예: Server#2 서버가 Server#1 서버 보다 시간이 빠를 경우, Server#1 번의 시간으로 Server#2의 시스템시간을 동기화 할 경우, 동일한 시간대의 데이터가 생성되므로, 데이터 정합성에 오류가 발생한다. 이에 시간 동기화는 즉시 반영되는 것이 아니라, 조금씩 그 차이를 줄여나가는 방식이다. - 표 12 참조)

##### 2. 시스템 환경

NTP 시간을 제공해주는 서버를 "NTP 서버", NTP 서버로 시간 동기화를 요청하는 서버를 "Slave 서버"로 명명한다.

서버명 및 IP 주소

서버명	IP 주소	역할
Server#1	10.10.10.1	NTP 서버
Server#2	10.10.10.2	Slave 서버
Server#3	10.10.10.3	Slave 서버



### 3. 내부 NTP 서버

#### 1) 외부 NTP 서버와 시간 동기화

내부 NTP 서버 시스템 시간을 인터넷 표준시(예:time.bora.net)으로 설정한다.

① 실행서버

Server#1(IP: 10. 10. 10. 1)

② 시나리오

내부 NTP 서버 역할로 운용할 서버의 시간 동기화 대상을 외부 NTP(예:time.bora.net, time.kornet.net)으로 설정하고, 설정(레지스트리)값을 확인 후 동기화 작업을 수행한다.

③ 작업

첫번째, 외부 NTP 서버를 기준으로 시간동기화 설정을 한다.

\* 시작 -> cmd -> w32tm /config /syncfromflags:manual /manualpeerlist:time.microsoft.com /update

표 1: 예상결과

```
C:\W>w32tm /config /syncfromflags:manual /manualpeerlist:time.microsoft.com /update
명령이 성공적으로 완료되었습니다
```

두번째, Windows Time 서비스의 설정값(레지스트리) 확인

\* 시작 -> cmd -> w32tm /dumpreg /subota\_key:Parameters

표 2: 예상 결과

```
C:\W>w32tm /dumpreg /subota_key:Parameters
값 이름      값 종류      값 데이터
-----
ServiceMain  REG_SZ      SvchostEntry_W32Time
ServiceDll   REG_EXPAND_SZ C:\WINDOWS\system32\w32time.dll
NtpServer    REG_SZ      time.bora.net,0x9 <- 외부 NTP 서버주소
Type         REG_SZ      NTP                <- 외부 NTP 서버 사용
```

세번째, Windows Time 서비스 재시작

\* 시작 -> cmd -> net stop w32tm  
-> net start w32tm

표 3: 예상결과

```
C:\W>net stop w32time
Windows Time 서비스를 멈춥니다.
Windows Time 서비스를 잘 멈추었습니다.

C:\W>net start w32time
Windows Time 서비스를 시작합니다.
Windows Time 서비스가 잘 시작되었습니다.
```

네번째, 시간 동기화

\* 시작 -> cmd -> w32tm /resync

표 4: 예상결과

```
[정상]
C:\W>w32tm /resync
로컬 컴퓨터에 다시 동기화 명령을 보내는 중
```

명령이 성공적으로 완료되었습니다.

[오류] : 방화벽에서 외부 UDP 123 포트가 막혀 있을 경우 발생할 수 있다.

C:\W>w32tm /resync

동기화 명령 전송 - local computer...

사용 가능한 시간 데이터가 없어 컴퓨터가 동기화하지 못했습니다.

## 2) NTP 서비스 제공을 위한 구성 확인

내부 NTP 서버가 외부 NTP 서버를 참조하는지 확인한다.

### ① 실행서버

Server#1(IP: 10.10.10.1)

### ② 시나리오

내부 NTP 서버 역할로 운용할 서버의 서비스를 확인하여 Slave 서버에서 연결할 수 있도록 구성 값을 확인한다.

### ③ 작업

첫번째, Windows Time 서비스 구동 확인

\* 시작 -> cmd -> sc query w32time

표 5: 예상 결과

```
C:\W>sc query w32time
SERVICE_NAME : w32time
TYPE           : 20 WIN32_SHARE_PROCESS
STATE          : 4 RUNNING
                (STOPPABLE,NOT_PAUSABLE,ACCEPTS_SHUTDOWN)
WIN32_EXIT_CODE : 0 (0x0)
SERVICE_EXIT_CODE : 0 (0x0)
CHECKPOINT     : 0x0
WAIT_HINT      : 0x0
```

두번째, NTP 서비스 구동 확인

\* 시작 -> cmd -> netstat -ano | findstr 123

표 6: 예상 결과

```
C:\W>netstat -ano | findstr 123
UDP 0.0.0.0:123    *:*  1128
UDP  0.0.0.0:62123  *:*  1428
UDP  [::]:123       *:*  1128
```

## 4. Slave 서버

### 1) Slave 서버 환경 구성

내부 NTP 서버(IP: 10.10.10.1)를 기준으로 Slave 서버의 시스템 시간을 설정한다. 시간 차이가 너무 많을 경우 동기화가 되지 않으니, 수동으로 근접한 시간을 맞추고 동기화 할 수 있도록 한다.

### ① 실행서버

Server#2(IP: 10.10.10.2)

Server#3(IP: 10.10.10.3)



(각 서버별로 수행)

② 시나리오

시간 동기화 대상을 내부 NTP 서버(IP: 10.10.10.1)로 설정하고, 동기화 작업을 수행한다.

③ 작업

첫번째, 내부 NTP 서버(IP: 10.10.10.1)를 기준으로 시간동기화 설정을 한다.

\* 시작 -> cmd -> w32tm /config /syncfromflags:manual /manualpeerlist:10.10.10.1 /update

표 7: 예상결과

C:\>w32tm /config /syncfromflags:manual /manualpeerlist:10.10.10.1 /update  
명령이 성공적으로 완료되었습니다.

두번째, Windows Time 서비스의 설정값(레지스트리) 확인

\* 시작 -> cmd -> w32tm /dumpreg /subota\_key:Parameters

표 8: 예상 결과

C:\>w32tm /dumpreg /subota\_key:Parameters

값 이름	값 종류	값 데이터
ServiceMain	REG_SZ	SvchostEntry_W32Time
ServiceDll	REG_EXPAND_SZ	C:\WINDOWS\system32\w32time.dll
NtpServer	REG_SZ	10.10.10.1 <- 외부 NTP 서버주소
Type	REG_SZ	NTP <- 외부 NTP 서버 사용

세번째, Windows Time 서비스 재시작

\* 시작 -> cmd -> net stop w32tm  
-> net start w32tm

표 9: 예상결과

C:\>net stop w32time  
Windows Time 서비스를 멈춥니다.  
Windows Time 서비스를 잘 멈추었습니다.

C:\>net start w32time  
Windows Time 서비스를 시작합니다.  
Windows Time 서비스가 잘 시작되었습니다.

세번째, 시간 동기화

\* 시작 -> cmd -> w32tm /resync

표 10: 예상결과

C:\>w32tm /resync  
로컬 컴퓨터에 다시 동기화 명령을 보내는 중  
명령이 성공적으로 완료되었습니다.

2) NTP 서버와 시간차 확인

특정 서버(ex.NTP 서버)와 실행한 서버와의 시간차이를 확인한다.

① 실행서버

Server#2(IP: 10.10.10.2)

Server#3(IP:10.10.10.3)  
(각 서버별로 수행하여 확인할 수 있다.)

### ② 시나리오

특정서버(Master Server:10.10.10.1)와 Slave Server(10.10.10.2)시간 동기화 대상을 Master Server 설정하고, 동기화 작업을 수행한다.

### ③ 작업

첫번째, Master Server를 기준으로 시간동기화 설정을 한다.

\* 시작 → cmd → w32tm /stripchart /dataonly /computer:10.10.10.1  
비교대상 서버와 시간이 일치한다면, 하기와 유사한 결과값을 볼 수 있다.

표 11: 예상결과

```
C:\W> w32tm /stripchart /dataonly /computer:10.10.10.1
Tracking 10.10.10.1 [10.10.10.1].
The current time is 2012-04-18 오후 19:17:13 (local time).
19:17:13, +00.2676328s    -> 비교 대상 서버와 +00.2676328 초만큼 차이가 난다.
19:17:15, +00.2593851s
19:17:17, +00.2589499s
19:17:19, +00.2428931s
^C                        -> 중지하기 위해서는 "Ctrl + c" 를 누르면 된다.
+00.xxxxxxx(또는 -00.xxxxxxx)로 대상 NTP 서버와 차이나는 시간만큼 표기된다.
예) +120.2428931s -> +120 초 차이가 남
```

## 5. NTP 설정 후 동기화 요청 및 동기화 확인

표 12: Slave 서버 동기화 설정

**NTP 서버(IP:10.10.10.1) 서버를 시스템 시간으로 설정한다.**  
C:\W>w32tm /config /syncfromflags:manual /manualpeerlist:10.10.10.1/update  
명령이 성공적으로 완료되었습니다.

**Windows Time 서비스를 재기동한다.**  
C:\W>net stop w32time  
Windows Time 서비스를 멈춥니다..  
Windows Time 서비스를 잘 멈추었습니다.

C:\W>net start w32time  
Windows Time 서비스를 시작합니다..  
Windows Time 서비스가 잘 시작되었습니다.

**NTP 서버(IP:10.10.10.1)로 지정한 서버와 동기화를 요청한다.**  
C:\W>w32tm /resync  
로컬 컴퓨터에 다시 동기화 명령을 보내는 중  
명령이 성공적으로 완료되었습니다.

**NTP 서버(IP:10.10.10.1)와 실행 중인 서버와의 시간차이를 확인한다.**  
C:\W>w32tm /stripchart /dataonly /computer:10.10.10.1  
10.10.10.1[10.10.10.1:123] 추적 중  
현재 시간은 2012-04-19 오후 22:47:25 입니다.  
22:47:25, +23.2364090s  
22:47:27, +22.7004942s -> 조금씩이지만 차이가 줄어드는 것을 볼 수 있다.  
22:47:30, +22.1639462s

22:47:32, +21.6430236s

^C -> 중지하기 위해서는 "Ctrl + c" 를 누르면 된다.

## 6. 주기적인 시간 동기화

주기적으로 시간을 자동으로 동기화 하게 만드는데, 방법은 다음과 같다.

1) 시작 - 실행 - regedit (혹은 Winkey+R - regedit)

2) HKEY\_LOCAL\_MACHINE  
SYSTEM  
CurrentControlSet  
services  
W32Time  
TimeProviders  
NtpClient

3) SpecialPollInterval 이름을 더블 클릭

4) 10진수로 바꿔서 본인이 원하는 숫자를 입력.  
60 = 60초, 600 = 10분, 3600 = 1시간, 86400 = 1일  
초로 계산한다.

5) 시작 - 실행 - cmd (혹은 Winkey+R - cmd)

6) Windows Time 서비스 재기동  
net stop w32time 엔터  
net start w32time 엔터

7) Windows Time 서비스 재부팅 시 자동 실행  
sc config w32time start= auto  
sc triggerinfo w32time start/networkon stop/networkoff

## 7. 참고 사항

### 1) NTP Client 활성화

```
reg add "HKLM\SYSTEM\CurrentControlSet\Services\W32Time\TimeProviders\NtpClient" /v Enabled /t REG_DWORD /d 1 /f
```

### 2) 동기화 요청주기 설정(60초)

```
reg add "HKLM\SYSTEM\CurrentControlSet\Services\W32Time\TimeProviders\NtpClient" /v SpecialPollInterval /t REG_DWORD /d 60 /f
```

### 3) 동기화 조건 설정

```
reg add "HKLM\SYSTEM\CurrentControlSet\Services\W32Time\Config" /v MaxPosPhaseCorrection /t REG_DWORD /d 0xFFFFFFFF /f
reg add "HKLM\SYSTEM\CurrentControlSet\Services\W32Time\Config" /v MaxNegPhaseCorrection /t REG_DWORD /d 0xFFFFFFFF /f
```

기본 값은 54000초(15시간)인데, 현재 시간과 NTP 서버의 시간이 15시간 이상 차이가 나면 동기화 하지 않

는다. 이 값을 0xFFFFFFFF로 변경하면 시간 차이에 상관없이 무조건 동기화 하겠다는 뜻이다.

#### 4) NTP 서버 설정

```
w32tm /config /syncfromflags:manual /manualpeerlist:"kr.pool.ntp.org"
```

NTP 서버는 기본적으로 "time.windows.com"을 우리나라 pool time server인 "kr.pool.ntp.org"로 지정함.

#### 5) Windows Time 서비스가 자동으로 시작되도록 설정

```
sc config w32time start=auto
sc triggerinfo w32time start/networkon stop/networkoff
```

Windows Time 서비스는 기본적으로 Domain에 가입된 상태에서만 자동으로 시작되게 트리거 설정되어 있다. 따라서 이 트리거를 네트워크 시작/중지로 변경해 주는 것이다. 이렇게 하지 않으면 Workgroup 환경에서는 "sc config w32time start=auto" 해두어도 실제로는 리부팅 후 자동으로 서비스가 올라오지 않는다. 따라서 반드시 필요한 설정이라 할 수 있다.

#### 6) Windows Time 서비스 재시작

```
net stop w32time
net start w32time
```

#### 7) NTP 동기화 즉시 실행

```
w32tm /resync /nowait
```

#### 8) 현재 NTP 동기화 상태를 확인

```
w32tm /query /status
```

```
reg add "HKLM\SYSTEM\CurrentControlSet\Services\W32Time\TimeProviders\WNTpClient" /v Enabled /t REG_DWORD /d 1 /f
reg add "HKLM\SYSTEM\CurrentControlSet\Services\W32Time\TimeProviders\WNTpClient" /v SpecialPollInterval /t REG_DWORD /d 60 /f
reg add "HKLM\SYSTEM\CurrentControlSet\Services\W32Time\Config" /v MaxPosPhaseCorrection /t REG_DWORD /d 0xFFFFFFFF /f
reg add "HKLM\SYSTEM\CurrentControlSet\Services\W32Time\Config" /v MaxNegPhaseCorrection /t REG_DWORD /d 0xFFFFFFFF /f
w32tm /config /syncfromflags:manual /manualpeerlist:"kr.pool.ntp.org"
sc config w32time start=auto
sc triggerinfo w32time start/networkon stop/networkoff
net stop w32time
net start w32time
w32tm /resync /nowait
w32tm /query /status
```

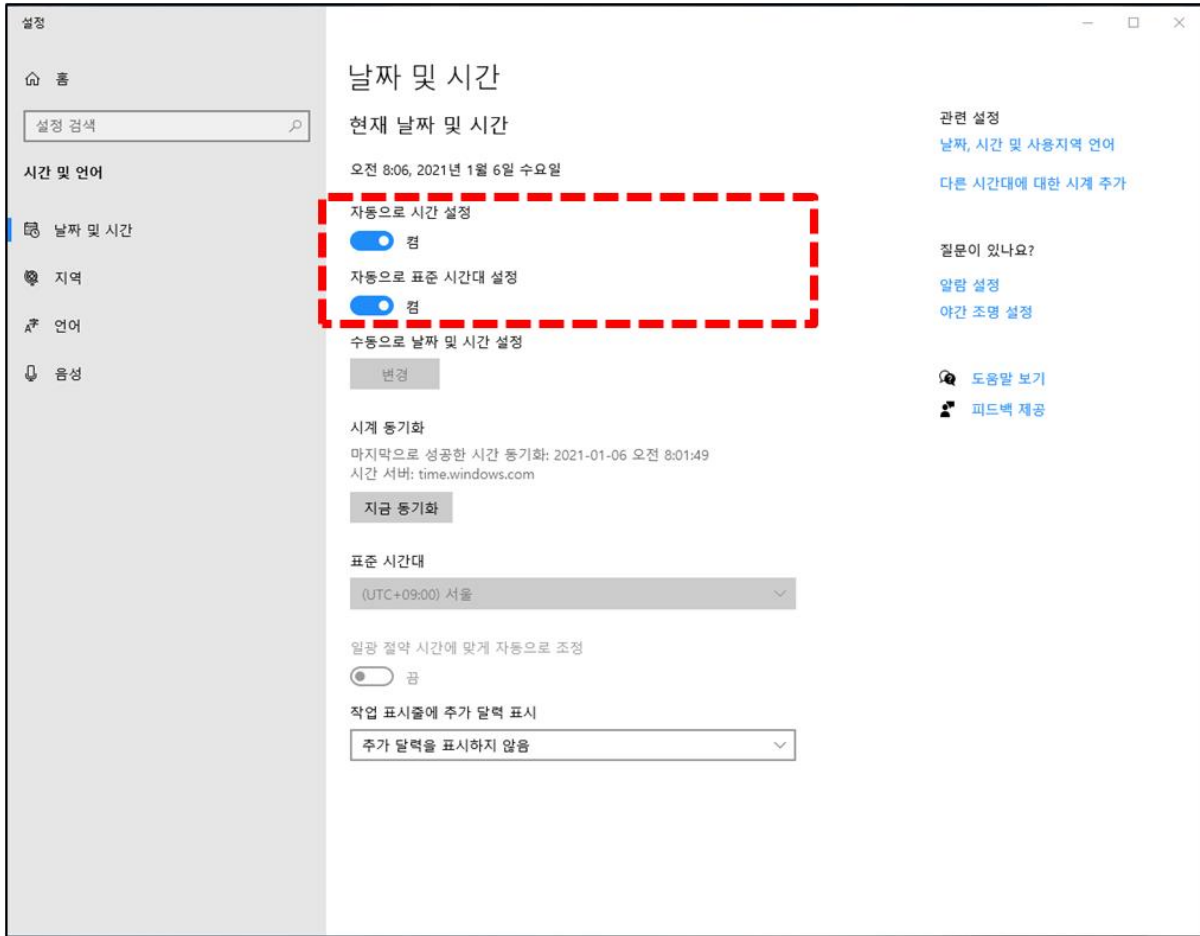
#### 9) Windows 현재 날짜 및 시간 확인 및 자동 설정



Windows 우측 하단의 날짜와 시간에 커서를 대고 "마우스 오른쪽 버튼"을 클릭하면 다음과 같은 팝업 화면이 나타난다.



위의 팝업 화면의 메뉴에서 "날짜/시간 조정(A)"을 클릭하면 다음과 같은 "날짜 및 시간"을 설정하는 화면이 나타난다.



위 화면의 항목 중 "자동으로 시간 설정"과 "자동으로 표준 시간대 설정" 항목을 "On"해야 한다.

## 4. About BaroPAM



Version 1.0 – Official Release – 2016.12.1  
Copyright © Nurit corp. All rights reserved.  
<http://www.nurit.co.kr>

제 조 사 : 주식회사 누리아이티  
등록번호 : 258-87-00901  
대표이사 : 이종일  
대표전화 : 02-2665-0119(영업문의/기술지원)  
이 메 일 : mc529@nurit.co.kr  
주 소 : 서울시 강서구 마곡중앙2로 15, 913호(마곡동, 마곡테크노타워2)