

## Multi-Tech iSMS - Kurzanleitung zur Einrichtung mit Nagios und Icinga

Der Multi-Tech iSMS™ ist ein dediziertes Gerät zum Versenden und Empfangen von SMS Textnachrichten. Man bindet das Gerät ganz einfach per Ethernetschnittstelle in das vorhandene Netzwerk ein und verbindet sich über das eingebaute Quad-Band GSM Modem mit dem GSM Netz. Jetzt kann jeder im Netzwerk SMS-Nachrichten versenden, was es Unternehmen ermöglicht Kunden, Abonnenten, Mitglieder und Interessenten via SMS mit einem Klick zu erreichen. Das API-Interface (siehe unten) erweitert die iSMS Funktionalitäten, indem es die Einbindung in andere Applikationen wie. z.B. Monitoring Server ermöglicht.

Herstellerlink: [http://www.multitech.com/en\\_US/PRODUCTS/Families/MultiModemiSMS/](http://www.multitech.com/en_US/PRODUCTS/Families/MultiModemiSMS/)



- ⊖ Wir empfehlen bei älteren Geräte, die mit der Firmware 1.50.7 ausgeliefert werden, ein Upgrade auf Version 1.51.9 durchzuführen. Hier wurden einige Fehler behoben. Die CLI via Telnet kann teilweise fehlerhaft arbeiten, sollte das Gerät Datenpakete von anderen Systemen erhalten. Die Funktionalität des Gerätes ist hiervon jedoch nicht beeinträchtigt.

### Einbindung in Nagios und Icinga

Für die Einbindung in Nagios und Icinga haben wir Integrationsskripte entwickelt.

Weitere Informationen und Downloadmöglichkeiten finden Sie auf unseren Community-Seiten:

[https://www.monitoringexchange.org/wiki/Plugin:check\\_smsfinder.pl](https://www.monitoringexchange.org/wiki/Plugin:check_smsfinder.pl)

<https://www.netways.org/projects/plugins/files>

### Disclaimer

Copyright © 2013 NETWAYS GmbH  
All rights reserved

⊖ Die NETWAYS GmbH haftet nicht für etwaige Fehler in dieser Dokumentation. Eine Haftung für mittelbare und unmittelbare Schäden, die im Zusammenhang mit der Lieferung oder dem Gebrauch dieser Dokumentation entstehen, ist ausgeschlossen, soweit dies gesetzlich möglich ist. Diese Dokumentation enthält urheberrechtlich geschützte Informationen. Alle Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie die Übersetzung, bleiben vorbehalten. Kein Teil der Dokumentation darf in irgendeiner Form (durch Fotokopie, Mikrofilm, oder ein anderes Verfahren) ohne vorherige schriftliche Zustimmung der NETWAYS GmbH reproduziert, oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

## Vorbereitung des iSMS

Die Grundkonfiguration (Anschluss/Netzwerk etc.) können Sie dem beigelegten Quick-Installation Guide entnehmen.

### Eingabe der Pinnummer:

⊖ Bitte deaktivieren Sie die Pinnummer auf Ihrer Simkarte. Dies können Sie mit jedem handelsüblichen Handy durchführen.

Wenn Sie die Simkarte bereits im SMSFinder eingebaut haben, können Sie dies auch per Telnet durchführen:

```
> telnet [IP-Adresse iSMS] 5000  
  
> AT+CPIN=1234  
OK  
> AT+CLCK="SC",0,"1234"  
OK
```

"1234" steht hier für die Pinnummer Ihrer Simkarte.

Sie können aber auch die Pinnummer wieder aktivieren:

```
> AT+CLCK="SC",1,"1234"  
OK
```

Den Status überprüft man mit

```
> AT+CLCK="SC",2
```

Dabei bedeutet

```
+CLCK: 0 - PIN-Abfrage ist deaktiviert  
+CLCK: 1 - PIN-Abfrage ist aktiviert
```

## Signalstärke

```
>at+csq
```

Ausgabe:

```
>+CSQ: 24,0  
>OK
```

**i** Erklärung:

0= -113dBm  
1 = -111dBm  
2 - 30= -109... -53dBm  
31= -51dBm+  
99= n/a bzw. "not detectable" - nur auswertbar während einer bestehenden Datenverbindung.  
Der Wert sollte im Ergebnis nicht unter -95dBm liegen.

## Test SMS versenden

```
>at+cmgs="+4916xxxxxxxxxx"  
>[Text der Testnachricht]  
>STRG+Z [ENTER]
```

Ausgabe:

```
+CMGS: 28  
  
OK
```

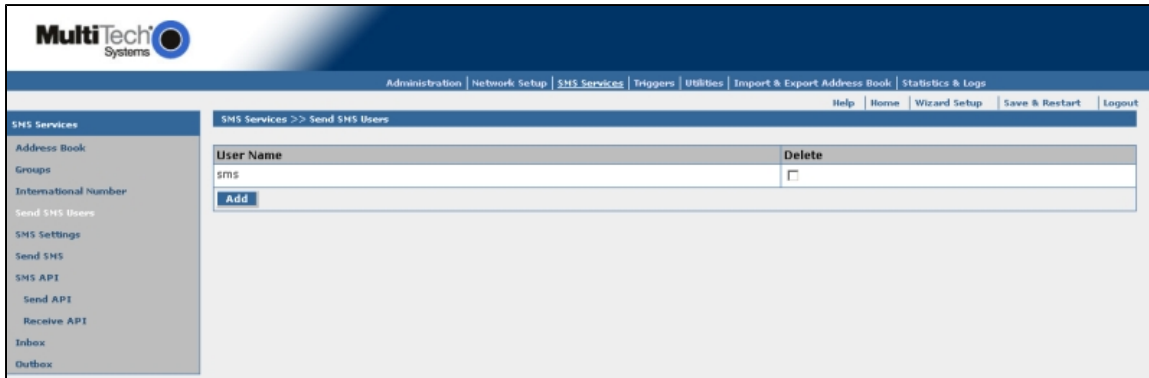
- i**
- Immer das +49 Format verwenden
  - Wenn das Modem in PDU Modus eingeschaltet ist, ist der Versand reiner Textnachrichten nicht möglich. Mit

```
AT+CMGF=1
```

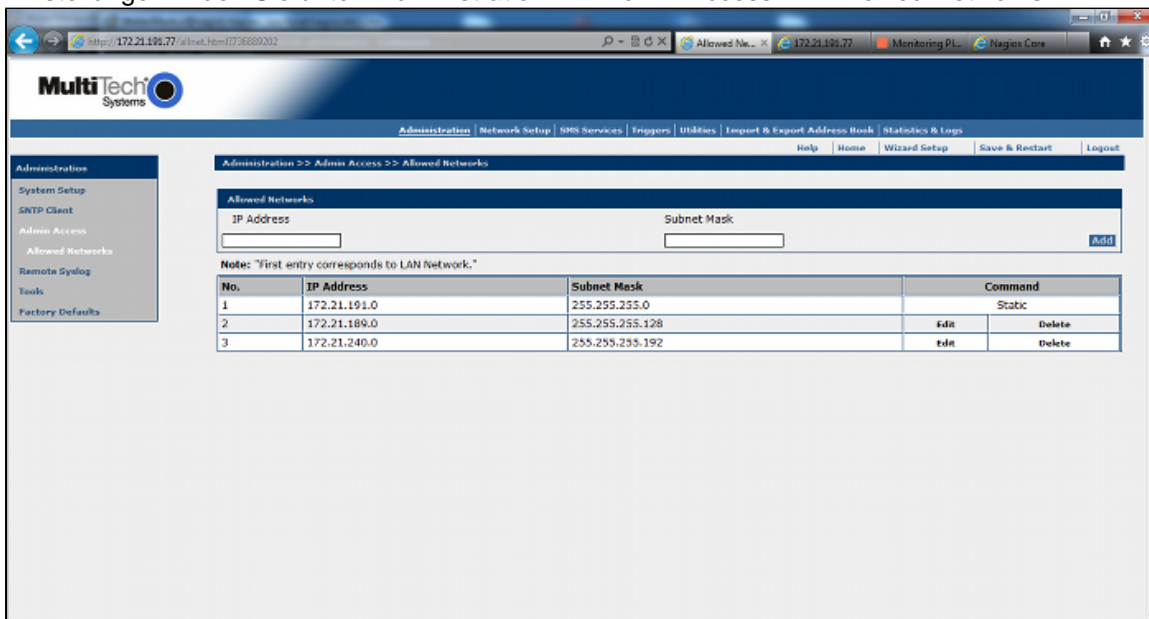
kann man in Text-Modus umschalten.

## SMS-Versand über das Webfrontend

Zum SMS-Versand sollten Sie noch einen entsprechenden User unter "SMS Services" --> "Send SMS Users" anlegen:



Zur Sicherheit können Sie die IP-Adressen, die auf den SMSFinder zugreifen dürfen, beschränken. Diese Einstellungen finden Sie unter "Administration" -> "Admin Access" -> "Allowed Networks".

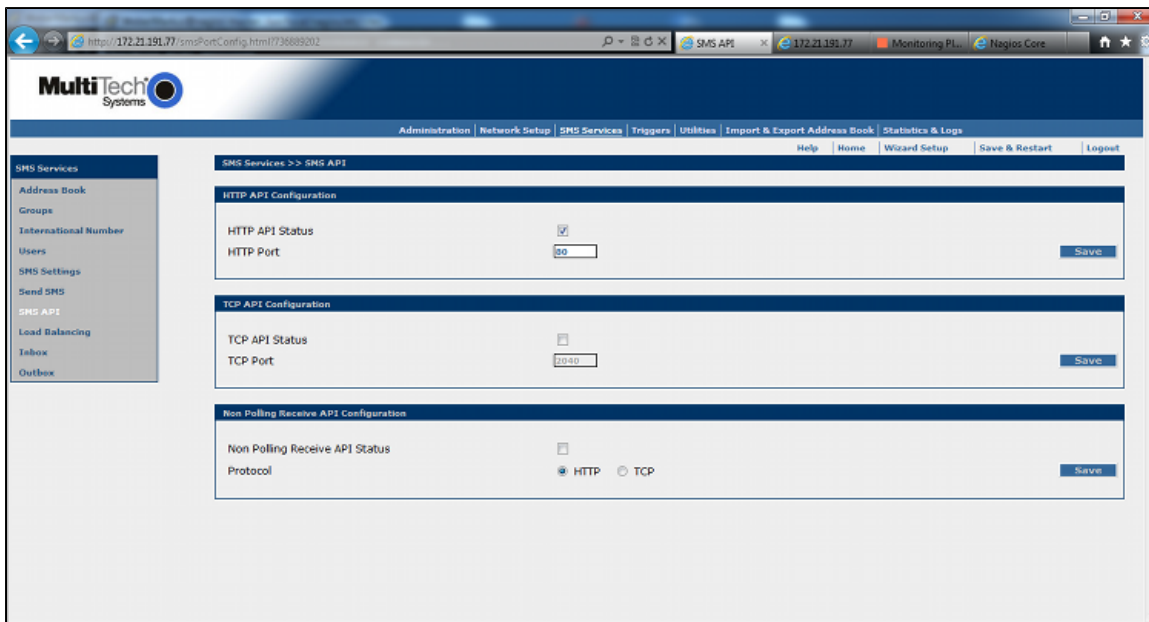


**⚠ Am Ende der Konfiguration bitte unbedingt die Einstellungen mit einem Klick auf "Save & Restart" im Flashspeicher des Geräts abspeichern!**

## Konfiguration der HTTP Send API für den SMS-Versand

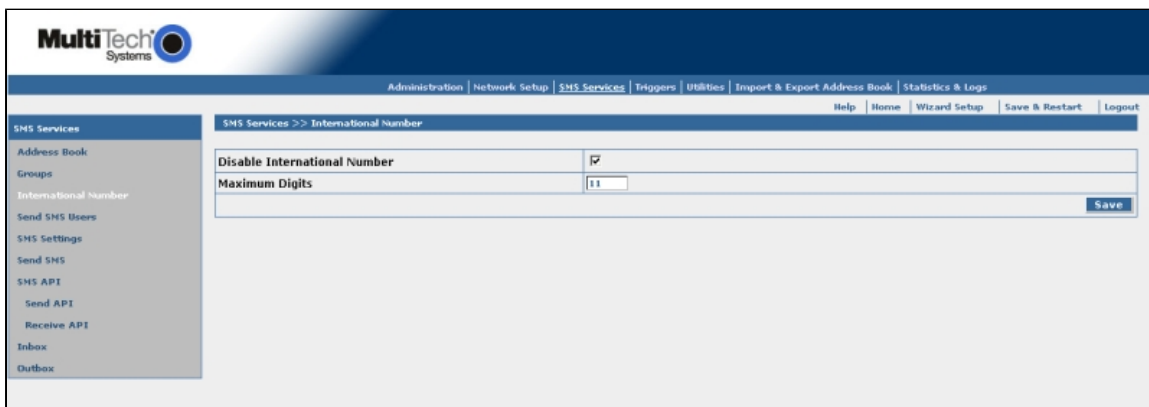
Der iSMS kann von Haus aus SMS über sein Webfrontend versenden. Damit man das Gerät in die Alarmierung mit einem Monitoring Server einbinden kann, muss man die HTTP Send API einrichten.

Diese Einstellungen kann man auf folgender Seite im Webfrontend unter "SMS Services" -> "Send API" vornehmen:



Hier einfach einen Haken bei "HTTP Send API Status" setzen.

Weiterhin sollten Sie die maximal Länge der Handynummer unter "SMS Services" --> "International Number" einstellen:

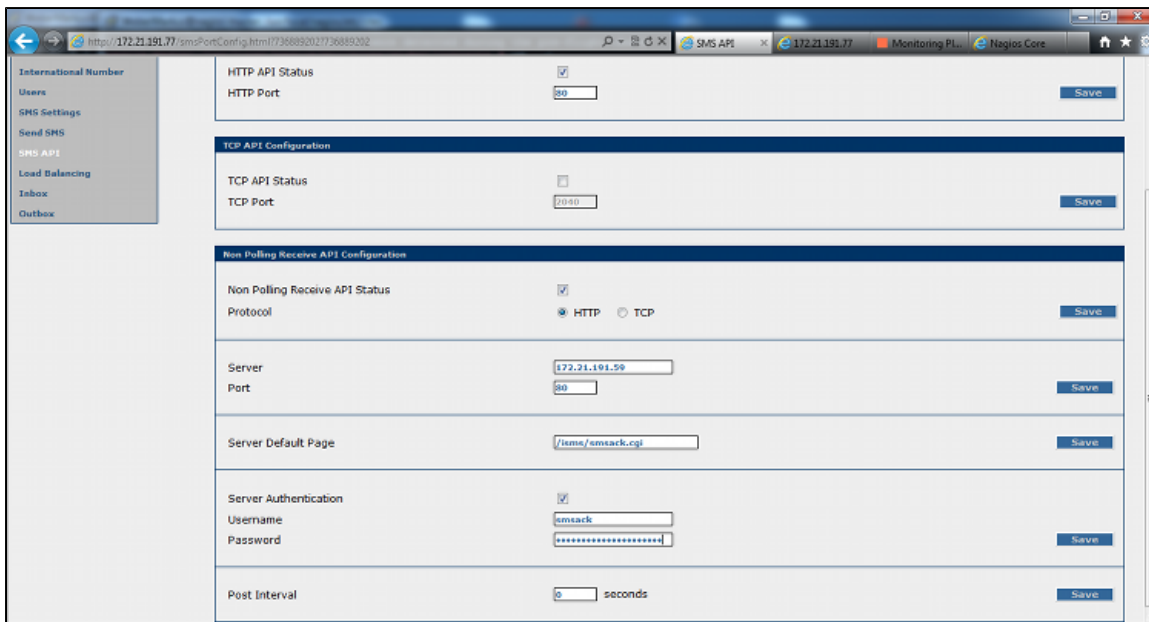


Wenn Sie den Haken bei "Disable International Number" entfernen, hat die Angabe der Nummernlänge keine weiteren Auswirkungen mehr.

**⚠ Am Ende der Konfiguration bitte unbedingt die Einstellungen mit einem Klick auf "Save & Restart" im Flashspeicher des Geräts abspeichern!**

## Konfiguration der HTTP Receive API für den SMS Empfang

Nachdem man unter "SMS Services" -> "Send API" die Einstellung "Non Polling Receive Status" aktiviert hat und abgespeichert hat, können nachfolgende Einstellungen vorgenommen werden:



Hier wird die IP-Adresse des Nagios- bzw. Icinga-Server, der Port und die URL des CGI angegeben. Ist diese URL Zugangsbeschränkt kann zusätzlich ein User mit Passwort angegeben werden.

**⚠ Am Ende der Konfiguration bitte unbedingt die Einstellungen mit einem Klick auf "Save & Restart" im Flashspeicher des Geräts abspeichern!**

## Grundsätzliche Funktion der HTTP Send API

Wenn Sie oben genannte Einstellungen am Gerät durchgeführt haben, können Sie mit einem einfachen HTTP Befehl eine SMS über den SMSFinder versenden.

Beispiel:

```
http://192.168.2.1:[portnumber]/sendmsg?user=user1&passwd=puser1&cat=1&to="7657867765"&text=message
```

Als Antwort sollte folgendes erscheinen:

**ID: 1**

Wobei die Zahl natürlich abweichen kann. Die angegebene Zahl wird vom iSMS nach Eingang der SMS vergeben.

Falls ein Fehler auftritt, gibt das Gerät einen der folgenden Fehlercodes zurück:

Error Code	Error Description	Response to
601	Authentication Failed	Send API, Query API
602	Parse Error	Send API, Query API

603	Invalid Category	Send API
604	SMS message size is greater than 160 chars	Send API
605	Recipient Overflow	Send API
606	Invalid Recipient	Query API
607	No Recipient	Send API
608	SMSFinder is busy, can't accept this request	Send API, Query API
609	Timeout waiting for a TCP API request	Send API
610	Unknown Action Trigger	Send API
611	Error in broadcast Trigger	Send API
612	System Error. Memory Allocation Failure	Send API, Query API

Sie können auch an mehrere Rufnummern, an einen Adressbucheintrag etc. SMS versenden. Dies ist im **Handbuch ab Seite 66** ausführlich beschrieben.

Einen Zeilenumbruch können Sie mit "%0A" generieren.

## Integration in Nagios bzw. Icinga

Nachfolgend wird davon ausgegangen, dass der iSMS tarball nach `/usr/local/nagios/libexec` entpackt wurde, damit finden sich unterhalb von `isms` alle nötigen Perlskripte.

## Überwachung des Modems

Im Folgenden ist ein Beispiel für ein Kommando und ein Service für die Überwachung des Modems zu finden.



```
# command definition to check SMSFinder via HTTP
define command {
    command_name        check_smsfinder
    command_line        $USER1$/isms/check_smsfinder.pl -H $HOSTADDRESS$ -u
                        $USER15$ -p $USER16$ -w $ARG1$ -c $ARG2$
}

# service definition to check the SMSFinder
define service {
    use                  generic-service
    host_name            smsfinder
    service_description  smsfinder
    check_command        check_smsfinder!40!20    # warning and critical in
percent
    ## maybe it's wise to alter the service/host template
    #contact_groups      smsfinders
}
```

## Benachrichtigung einrichten

Bevor Kontakte mit Pagernummern konfigurieren werden können, müssen noch die Kommandos für die Benachrichtigung angelegt werden.

```
define command {
    command_name        notify-host-by-sms
    command_line        /usr/local/nagios/libexec/isms/sendsms.pl \-H
                        $HOSTADDRESS:smsfinder$ \-u $USER13$ \-p $USER14$ \-n $CONTACTPAGER$ \-m
                        '$NOTIFICATIONTYPE$ $HOSTNAME$> is $HOSTSTATE$ /$SHORTDATETIME$/ $HOSTOUTPUT$'
}

define command {
    command_name        notify-service-by-sms
    command_line        /usr/local/nagios/libexec/isms/sendsms.pl -H
                        $HOSTADDRESS:smsfinder$ -u $USER13$ -p $USER14$ -n $CONTACTPAGER$ -m
                        '$NOTIFICATIONTYPE$ $HOSTNAME$, $SERVICEDESC$> is $SERVICESTATE$ /$SHORTDATETIME$/
                        $SERVICEOUTPUT$'
}
```

Sollten diese, so übernommen werden, ist zu beachten das ein Hostobjekt *smsfinder* existiert und in der *resource.cfg* mit *\$USER13\$* und *\$USER14\$*, der Benutzer und sein Passwort für die HTTP-API des Modems eingetragen sind.

```
# contact definition - maybe it's wise to alter the contact template
define contact {
    contact_name          smsfinder
    use                   generic-contact
    alias                 SMS Nagios Admin
    # send notifications via email and SMS
    service_notification_commands
notify-service-by-email,notify-service-by-sms
    host_notification_commands    notify-host-by-email,notify-host-by-sms
    email                    nagios@localhost
    pager                    01725555555          # alter this please!
}

# contact definition - maybe it's wise to alter the contact template
define contactgroup {
    contactgroup_name      smsfinders
    alias                  SMS Nagios Administrators
    members                smsfinder
}
```

## Konfiguration für den SMS Empfang

Damit eine eindeutige Zuordnung des gesendeten Problems erfolgen kann, wird aus Informationen des Datensatzes in der NDO/IDO ein 3 oder 4stelliger Wert errechnet und mit versandt.

Möchte man nun das Problem auf Acknowledge setzen, schreibt man eine Antwort bestehend aus der empfangenden Zahl einem Leerzeichen und einem *ACK*. Um das Problem wieder auf *OK* zu setzen, ist anstatt *ACK* ein *OK* einzufügen.

Hierfür ist der Apache des Nagis- bzw. Icinga-Server um folgenden Kontext zu erweitern:

```
ScriptAlias /nagios/isms "/usr/local/nagios/libexec/isms"
<Directory "/usr/local/nagios/libexec/isms">
    Options ExecCGI
    AllowOverride None
    Order allow,deny
    Allow from 192.168.2.1    # SMS Finder
</Directory>
```

Die *commands* für den Versand sind noch anzupassen, so dass eine ID aus der Datenbank heraus generiert werden kann:

```
define command {
command_name      notify-host-by-sms
command_line      /usr/local/nagios/libexec/isms/sendsms.pl -H
$HOSTADDRESS:smsfinder$ -u $USER13$ -p $USER14$ -n $CONTACTPAGER$ -m
'$NOTIFICATIONTYPE$ $HOSTNAME$> is $HOSTSTATE$ /$SHORTDATETIME$ / $HOSTOUTPUT$'
--use-db --hostname="$HOSTNAME$" --type=$NOTIFICATIONTYPE$
}

define command {
command_name      notify-service-by-sms
command_line      /usr/local/nagios/libexec/isms/sendsms.pl -H
$HOSTADDRESS:smsfinder$ -u $USER13$ -p $USER14$ -n $CONTACTPAGER$ -m
'$NOTIFICATIONTYPE$ $HOSTNAME$, $SERVICEDESC$> is $SERVICESTATE$ $SHORTDATETIME$
$SERVICEOUTPUT$' --use-db --hostname="$HOSTNAME$" --service="$SERVICEDESC$"
--type=$NOTIFICATIONTYPE$
}
```

- ⊖ Wir empfehlen das Feld "SIM-Number" frei zu lassen, das iSMS kann sonst in einigen Firmwares sehr viele unnötige, automatisierte SMS verschicken. Alternativ ist es sinnvoller einen NTP-Server z.B. "0.de.pool.ntp.org" in den Zeiteinstellungen anzugeben.

#### **i** sendsms Plugin

Mit der Firmware-Version 1.51.9 wurde in das iSMS die Möglichkeit eingebaut, empfangene Nachrichten in einer XML-Datei an den Server zu senden. Mit der aktuellen Version des sendsms.pl Scripts, führt dies jedoch zu einem PERL-Fehler. Im Webinterface unter dem Punkt "SMS Services" -> "SMS API" ist daher ganz am Ende der Seite die Anzahl der zu versendenden SMS innerhalb eines XML-Dokuments auf **1** zu setzen.