

New Voice Tool Concept

A	The Concept
B	Batch Job Concept
C	Post Job Concept
D	DDE Direct Data Exchange
E	E & M Special Connexion to PBX
F	
G	
H	
IJ	Initialize NVTINI.DBF, line startup parameter
K	startup parameter, cont.
L	
M	NVE.EXE run a program without any phone-line
N	
O	
PQ	
R	
S	Statistics
Sch	
St	
T	
U	
V	
W	Compiler Errors, System Errors
X-Z	

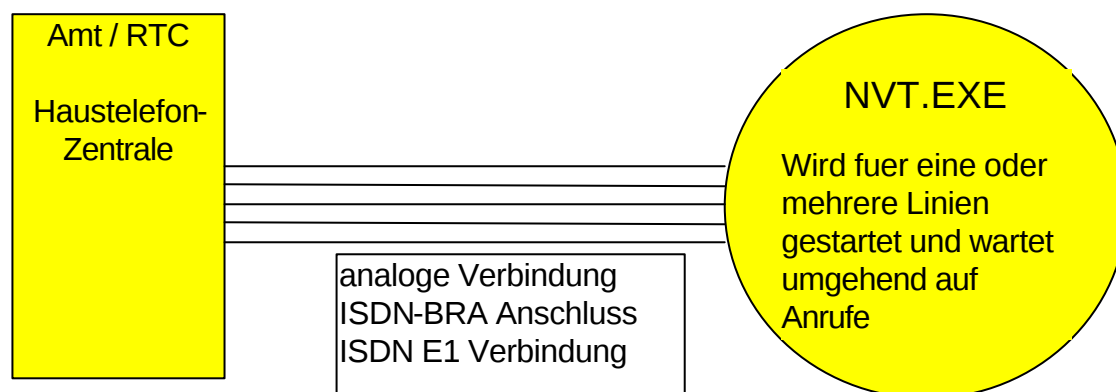
New Voice Designer

New Voice Tool - The Concept

Voice Applications

Das Konzept von New Voice Tool / New Voice Designer basiert auf drei elementaren Programmen und etlichen Zusatzprogrammen:

- NVT.EXE Programm zum Steuern der Telefoniekarten (Analog, ISDN PRI / BRA)
- NVD.EXE Programm zum Erstellen von Applikationen und Abläufen
- NVC.EXE Programm zum Kompilieren von erstellten Applikationen und Abläufen



Startkommandos:

[NVT] NVT.EXE < Liniennummer > < Programmname >

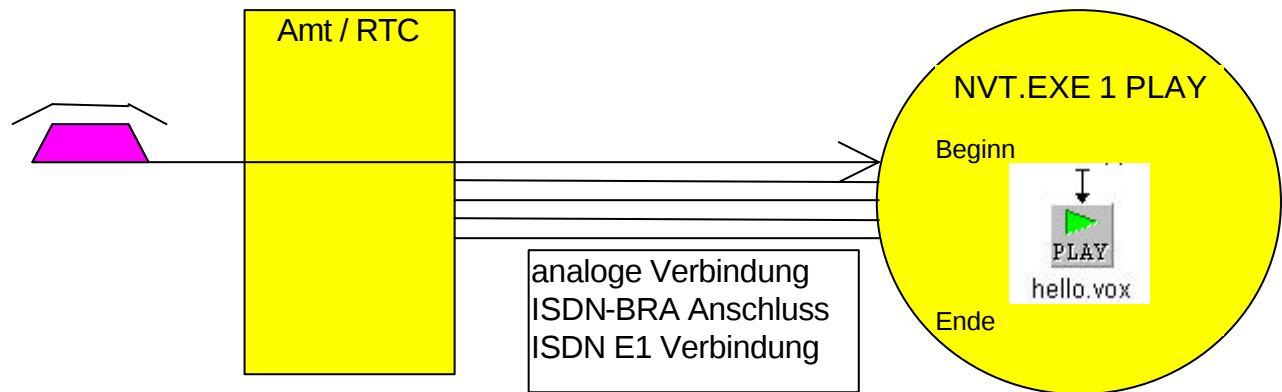
[NVT] NVT.EXE < erste Liniennummer > < letzte Liniennummer > < Programmname >

Demnach kann pro Linie ein Prozess gestartet werden, oder in einem Prozess mehrere Linien kontrolliert werden (ab Release 6.0).

Das Programm wird gestartet, sobald ein Anruf eintrifft. In diesem Programm wird dem System mitgeteilt, welche Ansagen gespielt werden und ob Nachrichten aufgezeichnet werden. Dazu gibt es eine Vielzahl an Befehlen, welche im folgenden beschrieben werden.

Dank einem integrierten Batchjob / Postjob Konzept kann jedem Prozess / jeder Linie unterschiedlichste Programme "in Auftrag" gegeben werden. Dies ist hilfreich bei ausgehenden Anrufen (Alarmierung / Weiterleitung, bei Fax Broadcast-Lösungen, etc.)

Beispiel eines Anrufs auf das System:



Voraussetzungen:

Gemäss der Struktur

[NVT] NVT.EXE < erste Liniennummer > < Programmname >
wurde die erste Linie mit dem Programm PLAY gestartet.

[NVT] NVT.EXE 1 PLAY

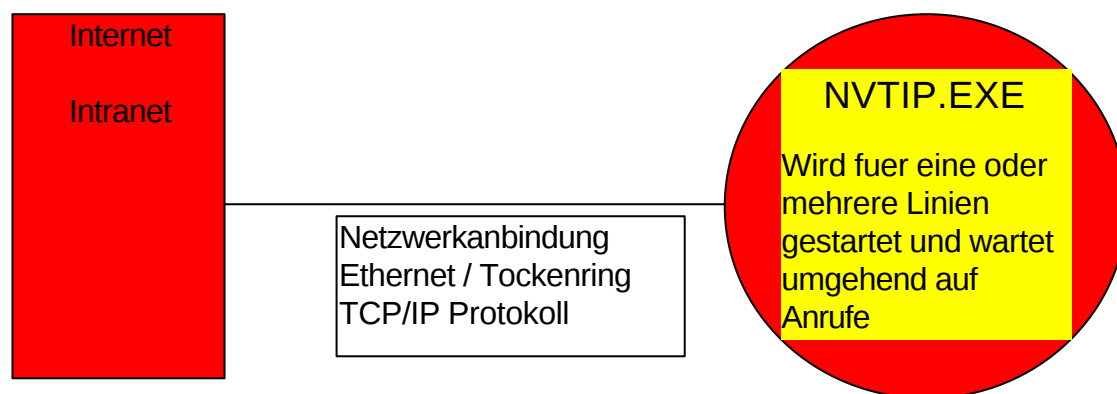
- Sobald der Anruf vom Telefon auf die erste Linie eintrifft, nimmt das System (NVT.EXE) den Anruf entgegen (nimmt den "Hörer" ab) und schaut im Programm PLAY nach, was zu tun ist.
- Das Programm PLAY besteht aus einem Icon, dem PLAY Icon:



- Demnach wurde im Program PLAY definiert, dass dem Anrufer die Datei "hello.vox" gespielt werden soll.
- Da nach diesem Befehl keine weiteren Anweisungen stehen, ist das Programm zu Ende und das System (NVT.EXE) beendet den Anruf (legt den "Hörer" auf).
- Das System (NVT.EXE) notiert die Dauer des Anrufs in Statistikdateien und wartet auf den nächsten Anruf auf dieser Linie.

Das Konzept von New Voice Tool / New Voice Designer basiert auf drei elementaren Programmen und etlichen Zusatzprogrammen:

- NVTIP.EXE Programm zum Steuern der IP-Karten zwecks New Voice over IP
-
- NVD.EXE Programm zum Erstellen von Applikationen und Abläufen
- NVC.EXE Programm zum Kompilieren von erstellten Applikationen und Abläufen



Startkommandos:

[NVT] NVTIP.EXE < Kanalnummer > < Programmname >

[NVT] NVTIP.EXE < erste Kanalnummer > < letzte Kanalnummer > < Programmname >

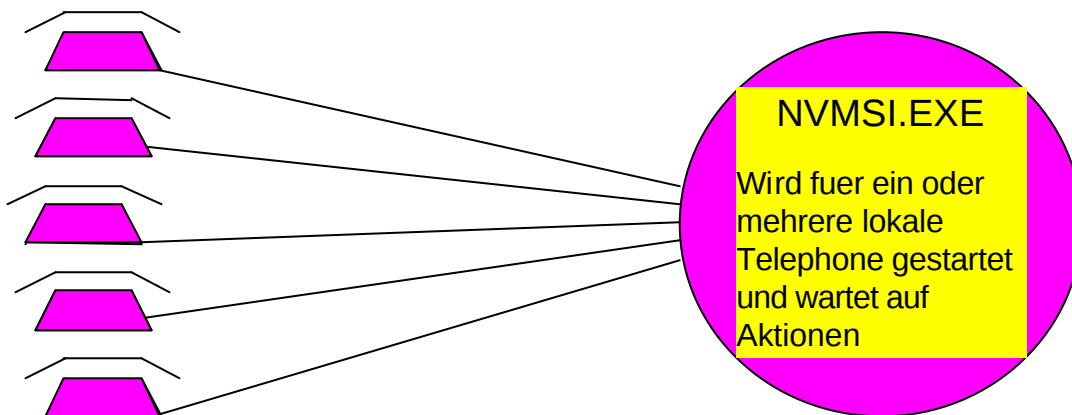
Demnach kann pro Kanal ein Prozess gestartet werden, oder in einem Prozess mehrere Kanäle kontrolliert werden (ab Release 6.0).

Das Programm wird gestartet, sobald ein Anruf eintrifft. In diesem Programm wird dem System mitgeteilt, wohin ein Anruf weitergeleitet werden soll. Dazu gibt es eine Vielzahl an Befehlen, welche im folgenden beschrieben werden.

Dank einem integrierten Batchjob / Postjob Konzept kann jedem Prozess / jedem Kanal unterschiedlichste Programme "in Auftrag" gegeben werden. Dies ist hilfreich bei ausgehenden Anrufen (Alarmierung / Weiterleitung auf PC-Stationen, etc.)

Das Konzept von New Voice Tool / New Voice Designer basiert auf drei elementaren Programmen und etlichen Zusatzprogrammen:

- NVTMSI.EXE Programm zum Steuern von lokalen Telefonen (Operating)
- NVD.EXE Programm zum Erstellen von Applikationen und Ablaeufen
- NVC.EXE Programm zum Kompilieren von erstellten Applikationen und Ablaeufen



Startkommandos:

[NVT] NVTMSI.EXE < Telefon > < Programmname >

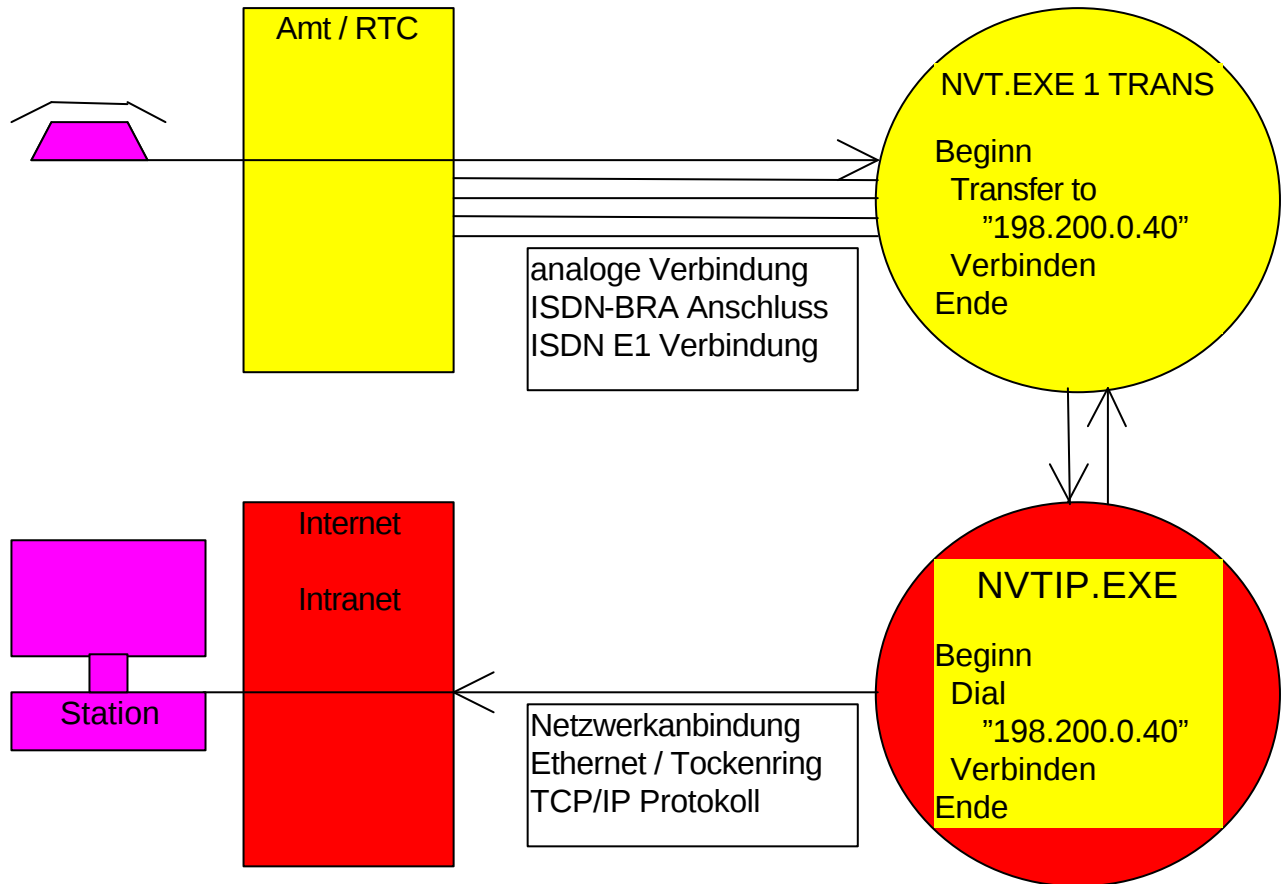
[NVT] NVTMSI.EXE < erstes Telefon > < letztes Telefon > < Programmname >

Demnach kann pro Telefon ein Prozess gestartet werden, oder in einem Prozess mehrere Telephone kontrolliert werden (ab Release 6.0).

Das Programm wird gestartet, sobald ein Anruf eintrifft. In diesem Programm wird dem System mitgeteilt, wohin ein Anruf weitergeleitet werden soll. Dazu gibt es eine Vielzahl an Befehlen, welche im folgenden beschrieben werden.

Dank einem integrierten Batchjob / Postjob Konzept kann jedem Prozess / jedem Telefon unterschiedlichste Programme "in Auftrag" gegeben werden. Dies ist hilfreich bei eingehenden Anrufen (von rufenden Telefonen oder PC-Stationen, etc.)

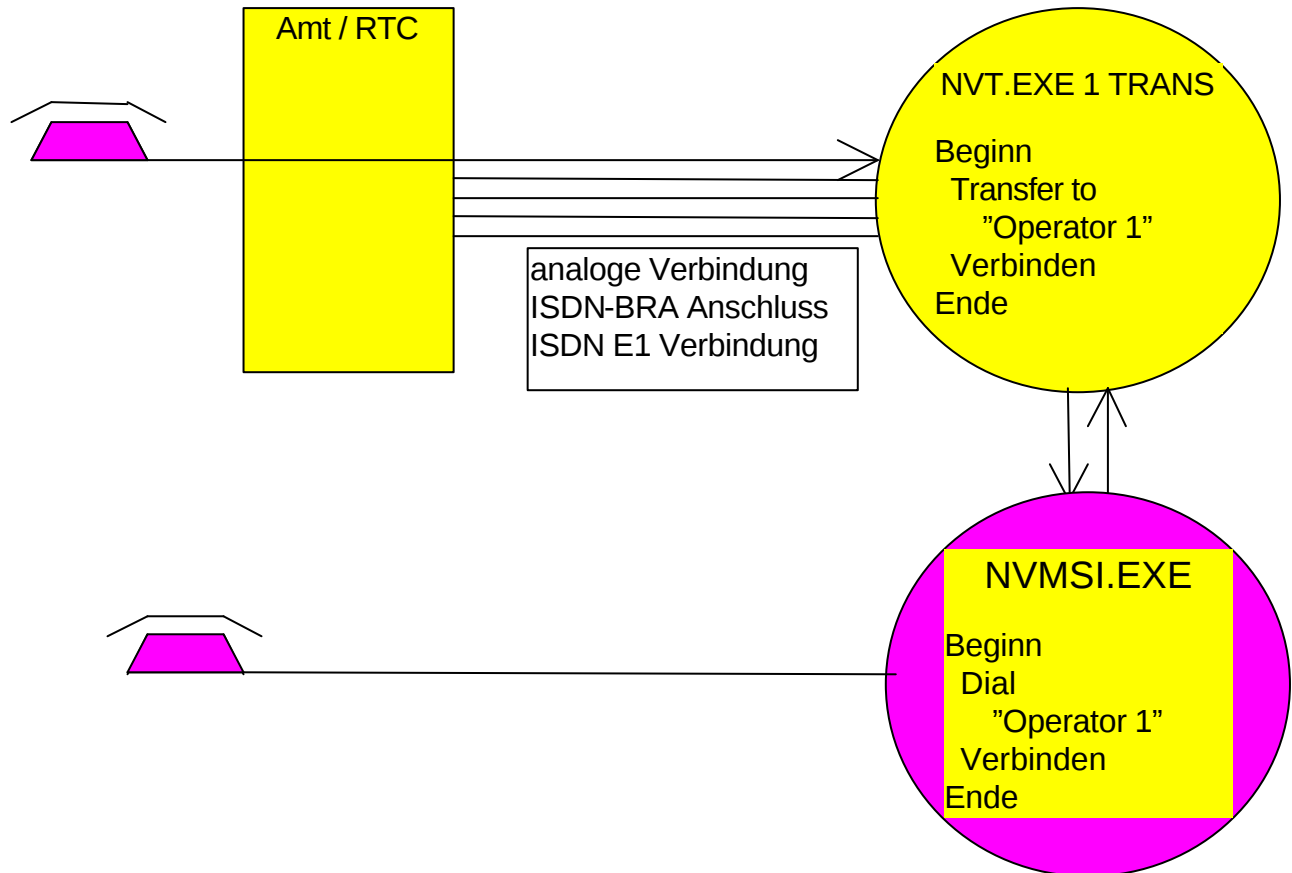
Beispiel einer Anrufweiterleitung ins Internet (New Voice over IP):



- Der Anruf kommt über das Telefonnetz auf das System (NVT.EXE) und wird von diesem Beantwortet.
- Das System (NVT.EXE) ermittelt anhand der vom Anrufer gewählten Nummer oder anhand der Anruferkennung oder anhand von Datum / Uhrzeit und aktiven Callcenter - Mitarbeitern die Ziel- IP- Adresse.
- Nun wird dem IP-Systemprozess (NVTIP.EXE) mitgeteilt, er solle ebendiese IP-Adresse anwählen.
- Sobald der Anruf beantwortet wurde, werden die Sprachkanäle durchgeschaltet - der Kunde und der Berater können miteinander sprechen.
- Sobald der Anrufer oder der Berater das Gespräch beendet, beenden auch die Systemprogramme (NVT.EXE und NVTIP.EXE) deren Programme und warten erneut auf den nächsten Anruf respektive auf den nächsten Auftrag zum Herauswählen.

Diese Beschreibung ist eine vereinfachte Darstellung der Abläufe.

Beispiel einer Anrufweiterleitung auf ein lokales Telefon:



- Der Anruf kommt über das Telefonnetz auf das System (NVT.EXE) und wird von diesem Beantwortet.
- Das System (NVT.EXE) ermittelt anhand der vom Anrufer gewählten Nummer oder anhand der Anruferkennung oder anhand von Datum / Uhrzeit und aktiven Callcenter - Mitarbeitern das Ziel-Telefon.
- Nun wird dem MSI-Systemprozess (NVTMSI.EXE) mitgeteilt, er solle ebendieses Ziel-Telefon anwählen.
- Sobald der Anruf beantwortet wurde, werden die Sprachkanäle durchgeschaltet - der Kunde und der Berater können miteinander sprechen.
- Sobald der Anrufer oder der Berater das Gespräch beendet, beenden auch die Systemprogramme (NVT.EXE und NVTMSI.EXE) deren Programme und warten erneut auf den nächsten Anruf respektive auf den nächsten Auftrag zum Herauswählen.

Diese Beschreibung ist eine vereinfachte Darstellung der Abläufe.

New Voice Tool - The Commands

Auflistung der wichtigsten Befehle und Befehlsgruppen

Play Voice-Commands	dial, play, rec, voldet, waitring,..
Speak Function	speak -string, -amount, -date, -time
Text-To-Speech Conversion	speak -tts
Voice Recognition	recognize
Conference Functions	conference
String Functions	strlen, strpart, strstr, strexchange
Mathematic Functions	+, -, *, /, ...
Time Functions	sleep, get time, get date, -difftime
Check Schedule Plan	checkschedule, checkweek
Fax Functions	sendfax, recfax
Goto a Sub-Program	do, proc, process, run
Com Port Command	setdtr, getdtr
Integration PABX	integration
Voice Mail Functions	getmbx, playmbx, recmbx, setmbx
Voice Mail Power Functions	playmbx, recmbx
File Functions -Ascii	fopen, fwrite, findfile
Data Base	dbuse, dbget, dbput...
ODBC Commands	odbcuse, odbcselect, odbcsql, odbcpout...
Pipe Commands	openpipe, putpipe, sendpipe closepipe...
Socket Commands	opensocket, putsocket, sendsocket closesocket.
Random Generator	random

Die meisten Befehle wurden auf Kundenwunsch implementiert. Somit sind diese Funktionen sehr praxisnah und sehr einfach zu bedienen. Benötigen Sie spezielle Funktionen und Befehle, teilen Sie uns dies mit, wir haben bestimmt eine Lösung für Sie bereit.

batch job concept

broadcast messages

Name:	nvtbatch.exe	batch-job interface
	nvtque.exe	batch-job utilites post-job utilities
	nvtqmngr.exe	batch-job monitor
Category:	Batch - Library	New Voice Tool - Swiss

r Description

To send broadcast voice-/fax-messages, to call operators or to set message lamps different functions and concepts are proposed by your New Voice Tool. The batch-job concept is line oriented and needs to have the queue manager nvtqmngr.exe started.

The post-job concept is an easy, fast and reliable alternative to the batch-job concept.

r Note

To get more information about every command, being part of any program and library, start the program without parameter. All modules are subject to evaluate in the future.

nvtque.exe help

NVT Queue-Commands V1.16 help

use: nvtque command option1 option2 option3

the NVT Queue-Commands are:

use: append first_line programname option1 option2 option3 option 4

use: batchpost line programname option1 option2 option3 option4

:

nvtbatch.exe

Usage: nvtbatch.exe [options] from_num [to_num] [field_name new_value]

from_num view/modify phone number from_num

to_num if present: view/modify from from_num to to_num

field_name if present: modify field_name with new_value

r Extendend functions

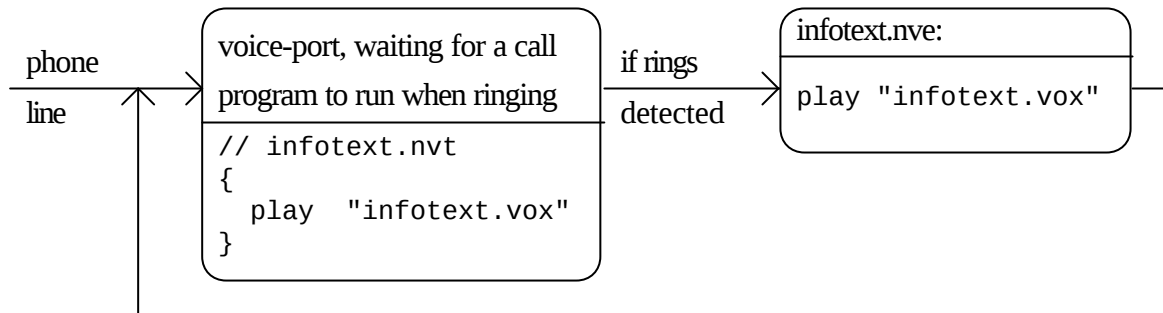
In this document you'll find also description about post-jobs (not line related batch-job) and the execution of any kind of batch-jobs by nve.exe

batch job programming `nvt.exe / nvtbatch.dbf` run a program on a free line

Description

Without Batch-Job:

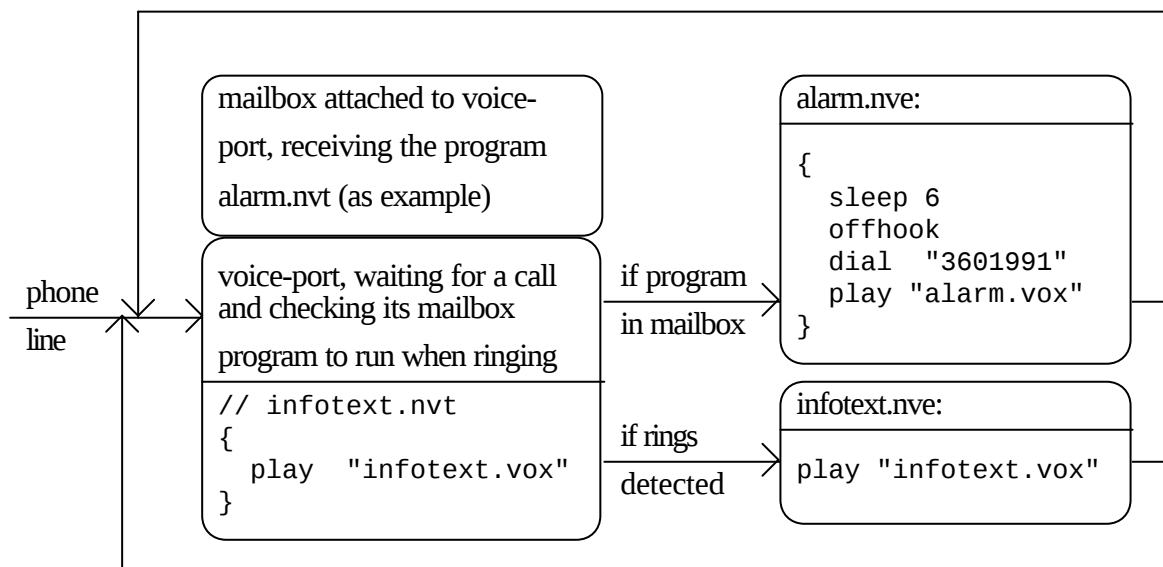
`nvtbatch.dbf` is absent



The voice-port / nvt-process ist waiting for a call and can not be used for other tasks.

With Batch-Job:

`nvtbatch.dbf` is present



The voice-port / nvt-process ist waiting for a call and can be used for other tasks. Every voice-port checks its mailbox every 10 or 20 seconds while waiting for rings.

Note: check also the fast post-job concept for alarming- and helpdesk-concepts.

Note: use `nvtque.exe` to enable or disable a line from supporting batch-job.

batch job programming `nvt.exe / nvtbatch.dbf` run a program on a free line

Name:	<code>nvtbatch.exe <line_nr> progname <name_of_program></code>	
Inputs:	<code>line_nr</code>	defines the line to activate
	<code>progname</code>	field-name in <code>nvtbatch.dbf</code>
	<code>name_of_program</code>	the program to launch
Returns:	<code>progname = name_of_program</code>	the batch-job is launched the program runs only once
Includes:	<code>nvtbatch.exe</code>	modify dbase <code>nvtbatch.dbf</code>
Category:	Batch - Library	New Voice Tool - Swiss

r Description

In general, a voice-port is waiting for a call all day long. If a caller is calling, vox-files are played and messages are recorded, as described in the program passed to the port of New Voice Tool when launched.

Watchdog-Functions usually starts outgoing calls, to inform the system-operator, that the Harddisk is nearly full, or that a message as been left for him. As it is too expensive to dedicate one port only for operator-alerts or for outgoing calls, now EVERY line can be used to make outgoing calls between waiting for calls!

Simply depose the name of the program to launch into the database record of `nvtbatch.dbf` corresponding to the voice-port to use. Immagine this record as the mailbox of the voice-port. Every voice-port is checking its mailbox every 10 or 20 seconds.

If nobody calls the voice-port, and a program is in the mailbox, this program will be executed.

r Note

NVTBATCH.DBF has the same structure as NVTINI.DBF, so every parameter can be set individually for the batch-program. The parameter 'rings' is set to 0 for immediate start.

r Example

`[c:\nvt]nvtbatch 1 progname nvtalarm` (sets progname of line 1 of `nvtbatch.dbf` to `nvtalarm`)

```
// program nvtalarm.nvt to call the operator
{
    sleep 6           // wait 0.6 seconds before going offhook
    offhook
    dial "013601991" // dial the number of the operator
    if st= 1          // if operator answers the call
        play "nvtalarm.vox" // "Harddisk of system 5 nearly full"
    fi
}
```

batch job queueing `nvtque.exe / nvtqueue.dbf` queue a job on a queue

Name:	<code>nvtque.exe</code>	<code>append <line_nr> <program> [options]</code>
Inputs:	<code>line_nr</code>	defines the line to activate 0: can run on all lines
Options:	<code>program</code> <code>-s1textinfo</code> <code>-s2textinfo</code> <code>-s3textinfo</code> <code>-s4textinfo</code> <code>-i112345</code> <code>-i254321</code> <code>-i31</code> <code>-i422222</code>	the program to launch textinfo stored in <code>nvtini_s1</code> textinfo stored in <code>nvtini_s2</code> textinfo stored in <code>nvtini_s3</code> textinfo stored in <code>nvtini_s4</code> integer stored in <code>nvtini_i1</code> integer stored in <code>nvtini_i2</code> integer stored in <code>nvtini_i3</code> integer stored in <code>nvtini_i4</code>
Returns:	posts the job on <code>nvtqueue</code>	the batch-job is queued <code>nvtquemn.exe</code> passes the job to the given line of <code>nvtbatch.dbf</code> to be launched
Includes:	<code>nvtque.exe</code> <code>nvtquemn.exe</code>	postes job in <code>nvtqueue.dbf</code> queue-manager to move the jobs into <code>nvtbatch.dbf</code>
Category:	Batch - Library	New Voice Tool - Swiss

r Description

The mailbox of each voice-port can stock one and only one batch-job at a time. If several jobs should be executed by one line, `nvtqueue.dbf` can be installed. The batch-jobs will no longer be posted directly in `nvtbatch.dbf` by using `nvtbatch.exe`, but by posting the batch-job in the batch-queue. `nvtque append` postes the batch-job into `nvtqueue.dbf`.

r Queue-Initialisation

The program **`nvtque.exe init <nbr_entries>`** will initialize the queue. The maximum number of entries is set to 40 (can be rised up to 500 and more).

r Note

It is recommended to setup the queue with the maximum number of entries (40 / 500 / more) to prevent the system of overwriting queued batch-jobs with newer batch-jobs.
Do not forget to start the queuemanager **`nvtquemn.exe`** which dispatches the jobs.

batch job posting nvtque.exe / nvtqueue.dbf

post a queued job

Name:	nvtque.exe	batchpost <line_nr> <program> [options]
Inputs:	line_nr program	defines the destination line the program to launch
Options:	-s1textinfo -s2textinfo -s3textinfo -s4textinfo -i112345 -i254321 -i31 -i422222	textinfo stored in nvtini_s1 textinfo stored in nvtini_s2 textinfo stored in nvtini_s3 textinfo stored in nvtini_s4 integer stored in nvtini_i1 integer stored in nvtini_i2 integer stored in nvtini_i3 integer stored in nvtini_i4
Returns:	posts the job in nvtbatch	the batch-job is posted in nvtbatch.dbf. If the line- mailbox is allready occupied, the job returns on the queue nvtqueue.dbf
Includes:	nvtque.exe nvtquemn.exe	postes job in nvtbatch.dbf queue-manager to call nvtque batchpost
Category:	Batch - Library	New Voice Tool - Swiss

r Description

The mailbox of each voice-port can stock one and only one batch-job at a time. The function nvtque(batchpost) checks, wether the mailbox of the dedicated line is free. If so, the job is posted, with the program-name and its options s1-s4 and i1-i4. If the mailbox is allready occupied, the job is posted on the queue nvtqueue.dbf. Queued jobs are dispatched by the queue-manager nvtquemn.exe.

r Note

Please read the description of the queue-initialisation nvtque.exe (init) and the queue-manager nvtquemn.exe carefully.

batch job queue manager `nvtquemn.exe/nvtqueue.dbf` queue a job

Name:	<code>nvtquemn.exe <first_line> <last_line> <check_time></code>	
Inputs:	<code>first_line</code>	first line with mailbox
	<code>last_line</code>	last line with mailbox
	<code>check_time</code>	delta time to poste jobs in seconds (5 - 60 seconds)
Returns:	dispatches the jobs into <code>nvtbatch</code> from <code>nvtqueue</code> to <code>nvtbatch</code>	
Includes:	<code>nvtqueue.dbf</code>	queue-database
	<code>nvtbatch.dbf</code>	line-mailbox-database
	<code>nvtque.exe batchpost</code>	program to write all infos (programe, options) into <code>nvtbatch.dbf</code> in one call
Category:	Batch - Library	New Voice Tool - Swiss

r Description

The program `nvtquemn.exe` checks every 15 seconds, wether a batch-job has been posted in the queue `nvtqueue.dbf` by using the program `nvtque.exe` append.
 If a job has been posted, `nvtquemn.exe` will dispatch the batch-job into `nvtbatch.dbf` by using the program `nvtque.exe batchpost`.
 The job will stay in the queue, as long as the corresponding line-mailbox is occupied. If the queue is full, old entries will be overwritten by the new ones.

r Example

```
[c:\nvt]nvtque append 1 nvtalarm -s1013601991 (queues the alarm for operator 1)
[c:\nvt]nvtque append 2 nvtalarm -s1013601900 (queues the alarm for operator 2)
```

```
[c:\nvt]nvtquemn 1 8 20 (setup queue-manager for line 1 to 8, delay 20 seconds)
```

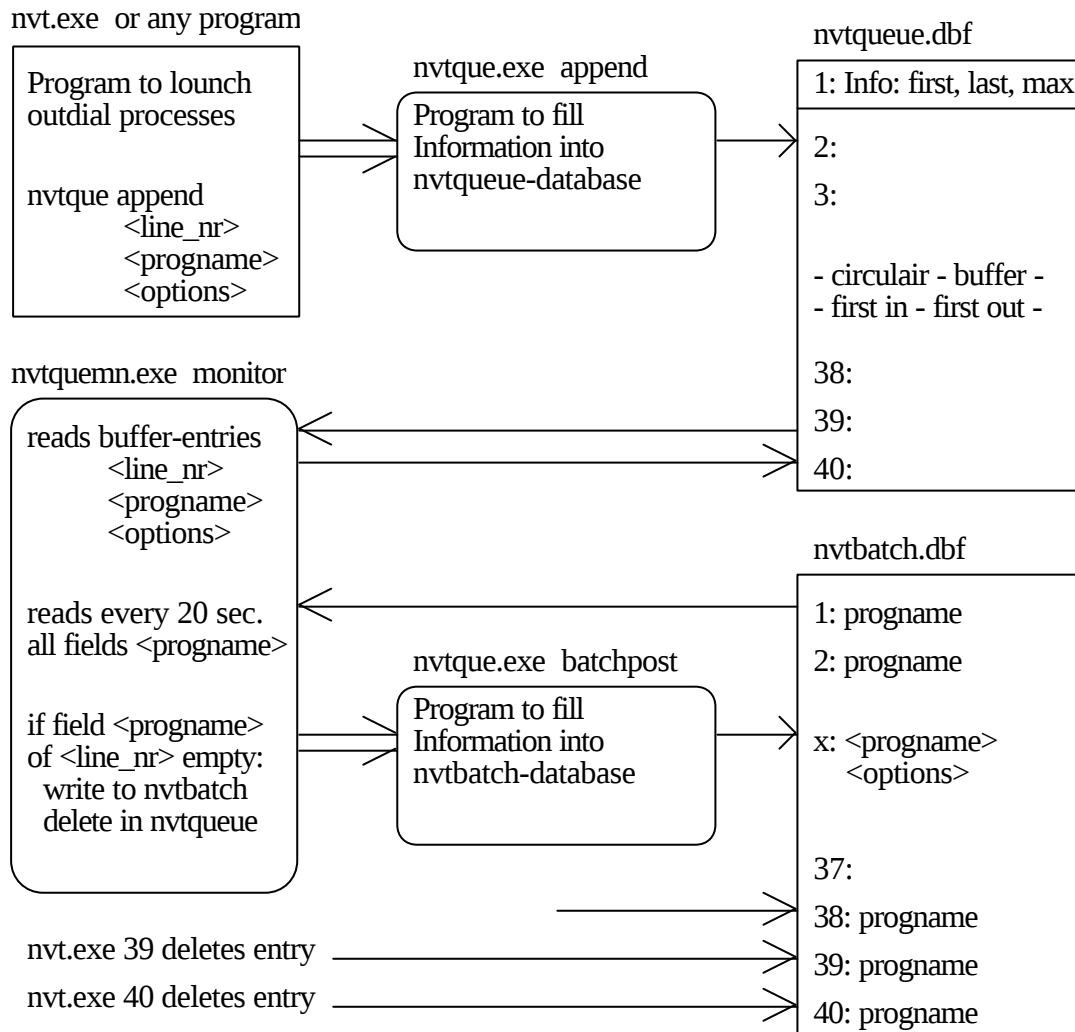
```
// program nvtalarm.nvt to call the operator
{
  sleep 6          // wait 0.6 seconds before going offhook
  offhook
  dial nvtini_s1    // dial the number of the operator
                  // line 1: 013601991, line 2: 013601900
  if st= 1          // if operator answers the call
    play "nvtalarm.vox" // "Harddisk of system 5 nearly full"
  fi
}
```

batch job queue manager nvtquemn.exe/nvtqueue.dbf dispatch batch-jobs

Category: Batch - Library

New Voice Tool - Swiss

r Concept - Design



r Flow Description

- Any program can start the program `nvtqueue.exe append` and poste a batch-job.
- `nvtqueue.exe append` puts the job onto the circulair buffer `nvtqueue.dbf`.
- If all entries are occupied, the last entry will be overwritten by the new entry.
- If `nvtqueue.dbf` is not initialized, use `nvtqueue.exe init 40` to initialize it.
- `nvtquemn.exe` must be started, to move the queued jobs into `nvtbatch.dbf`.
- `nvtqueue.exe batchpost` writes one job (`progrname` + `options`) into `nvtbatch.dbf`
- If the line-mailbox is occupied, the batch-job will be put onto the queue `nvtqueue.dbf`.
- Each line `nvt.exe <line_nr>` will check theyrs mailboxes and execute the batch-jobs.

r Commands - Overview

Setup NVTQUEUE MANAGER:

NVTQUEMN.EXE <first_line> <last_line> [delta_time]

Queue a batch-job with NVTQUE APPEND:

NVTQUE.EXE append <first_line> <prog_name> [-options]

-options: -S1str string to fill in nvtini_s1
 -S2str string to fill in nvtini_s2
 -S3str string to fill in nvtini_s3
 -S4str string to fill in nvtini_s4
 -I1int integer to fill in nvtini_i1
 -I2int integer to fill in nvtini_i2
 -I3int integer to fill in nvtini_i3
 -I4int integer to fill in nvtini_i4
 -future: -A ascending from Last_Line to <first_line>
 -R rotate, not linear

If <first_line> is equal 0, the job can be treated through every line supported by the queue-manager nvtquemn.exe <first_line> <last_line> <delay_sec>.

If no string is passed: "" is written into string and 0 is written into integer.

write a batch-job into NVTBATCH.DBF

NVTQUE.EXE batchpost <line> <prog_name> [-options]

-options: -S1str string to fill in nvtini_s1
 -S2str string to fill in nvtini_s2
 -S3str string to fill in nvtini_s3
 -S4str string to fill in nvtini_s4
 -I1int integer to fill in nvtini_i1
 -I2int integer to fill in nvtini_i2
 -I3int integer to fill in nvtini_i3
 -I4int integer to fill in nvtini_i4

initialize the queue NVTQUEUE.DBF

NVTQUE.EXE init <max_entries>

NVE.EXE batch job treatment nvtbatch.dbf nvtpost.dbf execute a job

Category: Batch - Library

New Voice Tool - Swiss

r Description

Sometimes it might be helpful to treat a postjob by nve.exe instead of nvt.exe.

Example: Any kind of simulation of outbound calls without really dialling a number.

r Example

NVE.EXE nvt_check_batch 1

This command starts the interpreter (without line interface). The filename containing _check_ forces nve.exe to scan nvtbatch.dbf for a batch-job (of line 1). If found, this job will be checked-out and executed by nve.exe

NVE.EXE nvt_check_post 1

This command starts the interpreter (without line interface). The filename containing _check_ forces nve.exe to scan nvtpost.dbf for any ,post-job' (note line related batch-job). If found, this job will be checked-out and executed by nve.exe

NVE.EXE nvt_check_post2 1

This command starts the interpreter (without line interface). The filename containing _check_ forces nve.exe to scan nvtpost2.dbf for any ,post-job' (note line related batch-job). If found, this job will be checked-out and executed by nve.exe

NVE.EXE nve_check_post 1

This command starts the interpreter (without line interface). The filename containing _check_ forces nve.exe to scan nvepost.dbf for any ,post-job' (note line related batch-job). If found, this job will be checked-out and executed by nve.exe

start nve execution nvtque.exe / nvtxxx.dbf launch nve.exe with options

Name:	nvtque.exe	execute <program> <line> <dbase> [options]
Inputs:	program	the program to launch
	line	defines the destination line
	dbase	location of nvtini database
Options:	-s1textinfo	textinfo stored in nvtini_s1
	-s2textinfo	textinfo stored in nvtini_s2
	-s3textinfo	textinfo stored in nvtini_s3
	-s4textinfo	textinfo stored in nvtini_s4
	-i112345	integer stored in nvtini_i1
	-i254321	integer stored in nvtini_i2
	-i31	integer stored in nvtini_i3
	-i422222	integer stored in nvtini_i4
Returns:	starts the nve.exe	the nvt program is launched
Includes:	nvtque.exe	postes job in nvtbatch.dbf
Category:	Basic Elements	

r Description

This function allows to execute a nvt program with nve.exe and the parameters i1 - i4 and s1 - s4. Usually, the database nveini.dbf is used to pass the parameters i1 - i4 and s1 - s4. It is also possible to use nvtini.dbf, nvtbatch.dbf, but be carefully.

As you can pass the whole path with the database-file-name, you can copy nvtini.dbf to another directory.

r Note

The integers nvtini_i1 - nvtini_i4 and strings nvtini_1 - nvtini_s4 are reset to "0" and " " if no value is passed.

Please note, that it's up to you to verify, that two calls of nvtque.exe execute will not overwrite each other. Therefore you have to select the linenumber in function of date and time or a counter.

r Note

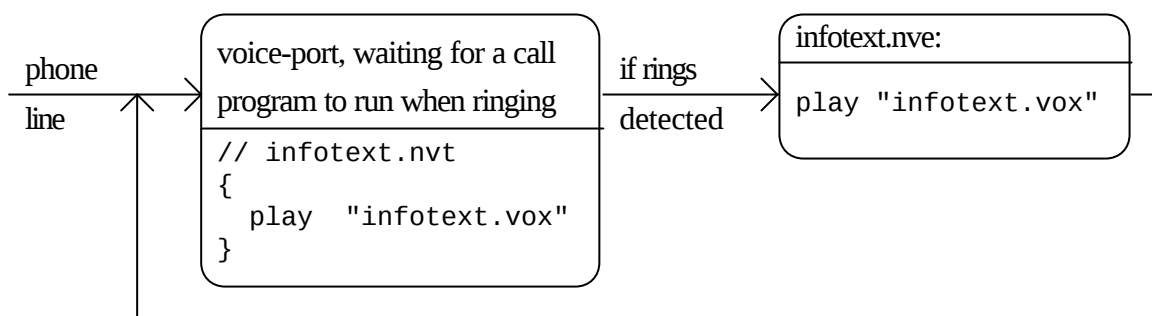
executepro: The program name is written into the field programname and deleted, after termination of the nvt-program. If two tasks launches a nvtque.exe executepro on the same line, the second will be set to the linenumber incremented by one (until one free line).

New Voice Post-Job

post job programming nvt.exe / nvtpost.dbf run a program on a free line

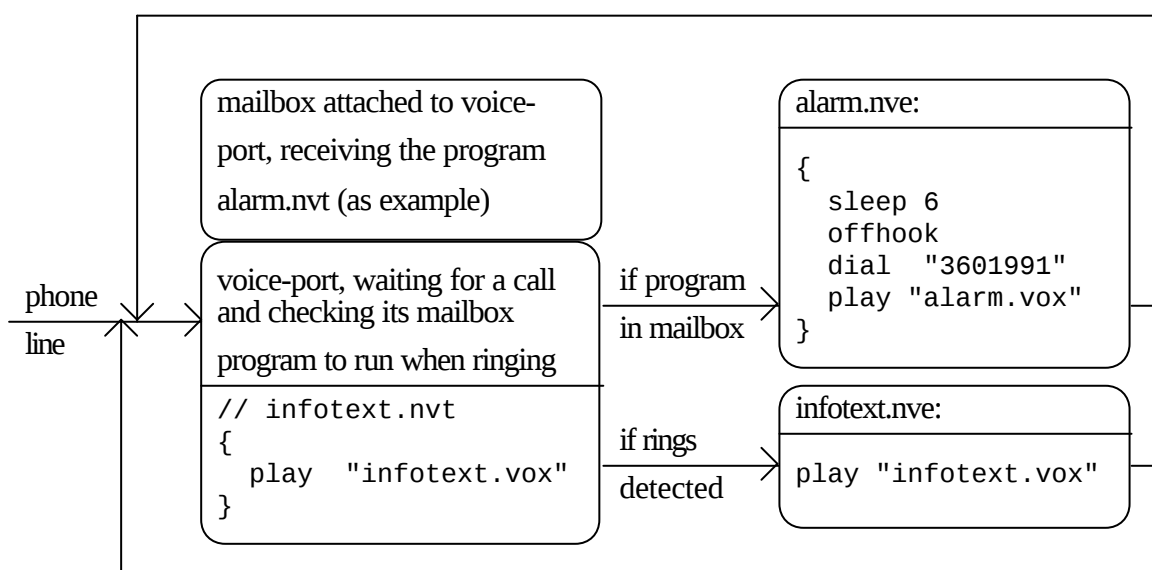
Description

Without Post-Job: nvtpost.dbf is absent or the line is disabled for post-job



The voice-port / nvt-process is waiting for a call and can not be used for other tasks.

With Post-Job: nvtpost.dbf is present and the line is enabled for post-job



The voice-port / nvt-process is waiting for a call and can be used for other tasks. Every voice-port checks the database nvtpost.dbf every 10 or 20 seconds while waiting for rings. If a job is posted, the line will execute the program.

Note: check also the batch-job concept for message-light-concepts.

Note: use nvtque.exe to enable or disable a line from supporting post-job.

post job queueing `nvtque.exe / nvtpost.dbf` queue a job on a queue

Name:	<code>nvtque postjob <line_nr> <program> [options]</code>	
Inputs:	<code>line_nr</code>	0: must be present, not used, reserved for future use
	<code>program</code>	the program to launch
Options:	<code>-s1textinfo</code>	textinfo stored in <code>nvtini_s1</code>
	<code>-s2textinfo</code>	textinfo stored in <code>nvtini_s2</code>
	<code>-s3textinfo</code>	textinfo stored in <code>nvtini_s3</code>
	<code>-s4textinfo</code>	textinfo stored in <code>nvtini_s4</code>
	<code>-i112345</code>	integer stored in <code>nvtini_i1</code>
	<code>-i254321</code>	integer stored in <code>nvtini_i2</code>
	<code>-i31</code>	integer stored in <code>nvtini_i3</code>
	<code>-i422222</code>	integer stored in <code>nvtini_i4</code>
Returns:	posts the job on <code>nvtpost.dbf</code>	the job is queued all lines having enabled the post-job option consult this queue to execute posted programs
Includes:	<code>nvtque.exe</code>	postes job in <code>nvtqueue.dbf</code>
Category:	Batch - Library	New Voice Tool - Swiss

❑ Description

This function posts a job onto the queue `nvtpost.dbf`.

❑ Queue-Initialisation

The program **`nvtque.exe postinit <nbr_entries>`** will initialize the queue. The base number of entries is set to 40 (can be rised up to 500 and more).

❑ Note

It is recommended to setup the queue with the maximum number of entries (40 / 500 / more) to prevent the system of overwriting queued jobs with newer jobs.
Make shure that at least one line has the post-job-fonctionnality enabled. Use `nvtque.exe` to change the state of a line.

```
// New Voice Tool Conference Demonstration Program      4.97
// Start Line 1 and Line 2 with the program divert.nvt
str cmd
int i
str fname
str line
int iline

proc waitforconnect{
    i= 20
    fname= "cnx"+nvtini_linenbr+ ".inf"
    fopen fname "write"
    fwrite ""
    fclose

    while i> 0
        fopen fname "read"
        fread cmd
        !"cmd: "+ cmd
        fclose
        //1xx: connect 2xx: busy 0xx: noanswer 5xx: error
        if 0= cmd
            getkey 1 -time 30 -cont
            if Hangup then
                !"caller went onhook"
                fname= "cnx"+ iline+ ".inf"
                fopen fname "write"
                fwrite "-1"
                fclose
                hangup
            else
                i= i-1
            fi
        else
            return
        fi
    wend
}
```

```

proc waitfordisconnect{
  i= 60
  fname= "cnx"+nvtini_linenbr+ ".inf"
  while i> 0
    fopen fname "read"
    fread cmd
    !"cmd: "+ cmd
    fclose
    if -1 # cmd
      getkey 1 -time 30 -cont
      if Hangup then
        !"caller went onhook"
        fname= "cnx"+ iline+ ".inf"
        fopen fname "write"
        fwrite "-1"
        fclose
        hangup
      else
        i= i-1
      fi
    else
      !"called personne disconnected"
      return
    fi
  wend
}
//=====MAIN
{
  play "hello.vox"
  cmd= "nvtque postjob 1 div_s1 -s1022161 -i1"+ nvtini_linenbr
  !cmd
  run cmd
  waitforconnect
  if i<= 0
    !"no connection" hangup
  fi
  iline= cmd
  if iline<= 100
    !"no answer" hangup
  fi
  if iline>= 200
    !"busy or error" hangup
  fi
  iline= iline- 100
  conference nvtini_linenbr iline 1
  waitfordisconnect
  conference nvtini_linenbr iline 0
}

```

```
str cmd
int i
str fname
str line
int iline
int state

proc waitfordisconnect{
  i= 60
  fname= "cnx"+nvtini_linenbr+ ".inf"
  while i> 0
    fopen fname "read"
    fread cmd
    !"cmd: "+ cmd
    fclose
    if -1# cmd
      getkey 1 -time 30 -cont
      if Hangup then
        !"caller went onhook"
        fname= "cnx"+ nvtini_i1+ ".inf"
        fopen fname "write"
        fwrite "-1"
        fclose
        hangup
      else
        i= i-1
      fi
    else
      return
    fi
  wend
}
```

conference

ISDN-E1 □ Example DIV_S1.NVT cont. (started from DIVERT.NVT)

```
//=====MAIN
{
  fname= "cnx"+nvtini_linenbr+ ".inf"
  fopen fname "write"
  fwrite ""
  fclose
  dial  nvtini_s1 -rings 6
  state= st
  !"call state: "+ state
  cmd= ""
  fopen fname "read"
  fread cmd
  !"cmd: "+ cmd
  fclose
  if -1= cmd
    !"initial caller disconnected"
    hangup
  fi

  if nvtini_linenbr> 9
    cmd= state+ nvtini_linenbr
  else
    cmd= state+ "0"+ nvtini_linenbr
  fi
  fname= "cnx"+ nvtini_i1+ ".inf"
  if state= 1
    fopen fname "write"
    fwrite cmd
    fclose
    !"start conference with "+ nvtini_i1
    conference nvtini_linenbr nvtini_i1 1
    waitfordisconnect
    conference nvtini_linenbr nvtini_i1 0
  fi
  if state= 2
    !"busy"
  fi
  if state= 0
    !"no answer"
  fi
  if state# 0 and state# 1 and state# 2
    !"error while dialling out"
  fi
  fopen fname "write"
  fwrite cmd
  fclose
}
```

New Voice DDE

NVTDDE_S.EXE

DDE-Support / Server W-NT

Name:	nvtdde_s.exe NVT	program name DDE- server name
Topic:	nve nvtini nvtbatch nvtque	new voice executer editor for nvtini.dbf editor for nvtbatch.dbf to launch batch-jobs batch-queue control
Includes:	nvtdde_c.exe nvtdde_s.exe nvtdde_e.cmd	specific exe-file specific exe-file launched by nvtdde_s.exe
Category:	Extended Interface - Commands	New Voice Tool - Swiss

☐ Description

New Voice Tool provides additional interface-functions based on the standard Direct Data Exchange (DDE). The commonly used function is, to start an outgoing call on a specific event raised in any program supporting DDE.

The Server-Name is NVT and the Topic includes as first parameter the name of the program to launch. All programs provided from New Voice Tool can be launched.

NVTDDE_S.EXE should be installed and run in the directory \nvt. It receives commands from NVTDDE_C.EXE or any other program beeing capable to send DDE commands.

☐ Start your NVT-DDE-Server

[nvt]\nvtdde_s.exe

A small window appears with the tilte New Voice DDE Server. The white lines indicates the number of nvtdde_s.exe launched on the system. If you start nvtdde_s.exe twice, every command will be launched twice. All command will be launched by using the command file nvtdde_e.cmd.

Every DDE-client message, received in the NVT-DDE-Server will be documented in the file \nvt\log\dde_11.log.

☐ Note

If you need other capabilities of NVT- DDE, please contact your NVT-representative.

NVTDDE_C.EXE

DDE-Support / Client W-NT

Name:	nvtdde_c.exe NVT	program name DDE- server name
Topic:	nve nvtini nvtbatch nvtque	new voice executer editor for nvtini.dbf editor for nvtbatch.dbf essentially to launch batch-jobs batch-queue control
Includes:	nvtdde_c.exe nvtdde_s.exe nvtdde_e.cmd	specific exe-file specific exe-file launched by nvtdde_s.exe
Category:	Extended Interface - Commands	New Voice Tool - Swiss

☐ Description

New Voice Tool provides additional interface-functions based on the standard Direct Data Exchange (DDE). The commonly used function is, to start an outgoing call on a specific event raised in any program supporting DDE.

The Server-Name is NVT and the Topic includes as first parameter the name of the program to launch. All programs provided from New Voice Tool can be launched. NVTDDE_S.EXE should be installed and run in the directory \nvt. It receives commands from NVTDDE_C.EXE or any other program being capable to send DDE commands.

☐ Start a Command through NVT-DDE

```
[nvt\nvtdde_c.exe nve calculat.nve 1
```

A small window appears with the title NVT-DDE then the command will be executed. Every DDE-client message, received in the NVT-DDE-Server will be documented in the file \nvt\log\dde_11.log. The command will be launched by using the command file nvtdde_e.cmd.

☐ Example: launch the batchjob dial_s1 on line 1

```
[nvt\nvtdde_c.exe nvtque batchpost 1 dial_s1 -s1111
```

☐ Note

If you want other capabilities of NVT- DDE, please contact your NVT-representative.

New Voice Initialization

startup - parameters

initializing / customizing

Name:	pmnvt.exe	line manager / configurator
	nvtini.exe	line configurator
	nvtbatch.exe	batch-job launcher
	nvtque.exe	batch-job utilites post-job utilities
	nvtqmgr.exe	batch-job monitor

Category: Setup - Library

New Voice Tool - Swiss

□ Description

The experience has shown, that it is usefull to pass several data to any voice-/ fax-application. As example, think about an application running on a system in Germany and one in France. Therefore it is very helpfull to have a variable containing the directory of all voice-prompts.

New Voice proposes about 200 predefined variables, which can be set on startup and modified during runtime.

See also „extended configuration flags“.

Line Startup Parameters

Use the pull-down menu **CONFIGURATION** to modify Startup Parameters. These parameters will be passed to the application-program on startup.

Note

The pull-down menu **BATCH** to modify Startup Parameters for broadcast voice-/fax-messages. These parameters will be passed to the application-program on startup.

Concept:

Programs very often should be adapted to client specific needs. Instead of modifying and recompiling programs, programs can change its behaviour in function of the content of global variables.

Imagine a program supporting vox-prompts in two different languages. Line 1 to 4 will startup with English and Line 5 to 8 will startup with French voice-prompts. Instead of writing two programs for the English and French version, the location of the vox-prompts can be passed in NVTINI_LG1.

Example:

```
'program    hello.nvt                                1. 1. 1995
{
    !"Wellcome to New Voice Tool"
    play "hello.vox"  nvtini_lg1
}
```

Now start the program `hello.nvt` with `nvtini_lg1= c:\gbvox\` for line 1 to 4 and `= c:\frvox\` for line 5 to 8

As at least in Europe, most of the systems will support two different languages, the variables NVTINI_LG1 and NVTINI_LG2 are introduced.

NVT proposes the database VM001 with 10'000 Mailboxes, located in any directory on any disk. NVT will find the database through the variable NVTINI_DB1, which is normally set to `c:\db1\`. To double the number of mailboxes, simply copy the database VM001 to another directory and change NVTINI_DB1 in your program to the alternate location.

Warning: do not forget to change also the directory of recorded messages, which is by default stored in `NVTINI_DBV1 = c:\dbv1\`

In this way, a lot of usefull variables are predefined by NVT. Please find a list of all variables on the following pages. Some preprinted pages will help you to note the specific values to set for a specific system.

Line Startup Parameters -cont

If you want the caller to change the language during the communication, simply program the following:

Example:

```
'program info.nvt                                1. 1. 1995
str voiceprompts
{
  'sets the location of the voiceprompts to nvtini_lg1
  voiceprompts= nvtini_lg1

  while key # "#"
    !"Wellcome to NVT, press 1 for info or 8 to change language"
    play "hello.vox" nvtini_lg1

    getkey key 1 -time 30
    if key= "1"
      play "info.vox" voiceprompts
    fi
    if key= "8" and voiceprompts= nvtini_lg1
      voiceprompts= nvtini_lg2
    fi
    if key= "8" and voiceprompts= nvtini_lg2
      voiceprompts= nvtini_lg1
    fi
  wend
}
```

Now start the program hello.nvt with nvtini_lg1= c:\gbvox\ for line 1 to 8 and nvtini_lg2= c:\frvox\ for line 1 to 8

Every caller will then hear the english wellcome message with the possability to change the language by pressing the key 8.

Line Startup Parameters -cont II

Parameter description for New Voice Tool applications

Use the pull-down menu **CONFIGURATION** to modify Startup Parameters. The variable-names used in this manual might differ to those indicated in the **CONFIGURATION** menu. All in all, these are the names of the system variables, which you can use in your NVT-application.

nvtini_rings	Number of rings detected, before going off hook, value 2 is recommended. 0: The program will start immediately, without waiting for a ring. The line will remain onhook, until the command OFFHOOK is applied. < 0: Number of rings detected, before starting the program. The line will remain onhook, until the command OFFHOOK is applied.
nvtini_maxrec	Defines the default value for maximal recording time of one message. This parameter can be changed with the option -TIME in the REC command.
nvtini_recqu	Defines the defaultvalue of the Quality of recording voiceprompts. 0: low quality; 3: best quality. Use 3 for prerecorded voiceprompts. Use 1 or 0 for messages recorded into mailboxes to optimize your diskspace.
nvtini_progname	Defines the name of the program at startup time. Do not append the extention to the name. Use 'demo' instead of 'demo.nvt'.
nvtini_prokdir	Defines the directory path, where the program is located at startup time. Example: 'c:\nvt \'
nvtini_statname	Defines the name of an userdefined statistic-file. Default is: 'statYYWW.log'. The following Letters will be exchanged at runtime: YYYY: 1995, YY: 95, WW: 52, MM: 12, DD: 31 DDD: 365
nvtini_statdir	Defines the directory of the userdefined statistic-file. Default is the local directory. We recommend to use c:\nvt\log
nvtini_maindir	Defines the directory, where common files are located, such as the standard statistic-file callYYMM.log and the file define.nvt.
nvtini_teldir	Defines the directory, where the monitor-files tel*.dat are located. For performance reasons store your tel*.dat files on a ram-disk.

Line Startup Parameters -cont III

Additional parameter description. Usefull with **New Voice Mail Power Functions**

- nvtini_i1.. i4** You can pass up to four integer values to your application at startup. You can modify these integers during a caller session and you'll find the modified values in your statistic-file defined in nvtini_statname.
- nvtini_s1.. s4** You can pass up to four strings to your application at startup. You can modify these strings during a caller session and you'll find the modified values in your statistic-file defined in nvtini_statname.
- nvtini_lg1** Defines the directory of the systemprompts of the first language. i.e. c:\dtvox\
- nvtini_lg2** Defines the directory of the systemprompts of the second language. i.e. c:\frvox\
- nvtini_dbv1** Defines the directory of the recorded voice-files-messages. i.e. c:\dbv1\
- nvtini_dbv2** Reserved for future use
- nvtini_db1** Defines the directory of the database VM001.DBF used by NVT - voicemail power functions. This database is also used by NVMS, New Voice Mail System. i.e. c:\db1\
- nvtini_db2** Reserved for future use
- nvtini_aab1** Defines the voice-file-prefix used by NVT - voicemail-power functions. This value is also used by NVMS. i.e. "VM" / "V"
Do not change this application abbreviation once the system is set up.
- nvtini_aab2** Reserved for future use
- nvtini_amt** Use "A" for a connection behind an analog trunk.
Use "C" for a connection to analog trunks with specific CH configuration
Use "I" for a connection to ISDN incoming lines
Use "Ab" for a connection to PABX with special Voicemail Integration
Use "E" for a Swissnet 2 connection using the ACULAB ETS300 board
Use "A1" for a connection to an analog trunk with special signalling level 1
Use "A2" for a connection to an analog trunk with special signalling level 2
Use "A3" for a connection to an analog trunk with special signalling level 3
Use "A4" for a connection to an analog trunk with special signalling level 4
- nvtini_din** Defines an Integer. Any value from -9999 to 29999 is possible.

Line Startup Parameters -cont III

Additional parameter description. Usefull with **New Voice Mail Power Functions**

nvtini_tt	Defines a phone or fax-number to dial
nvtini_ts1	Defines a phone or fax-number of your first secretary
nvtini_ts2	Defines a phone or fax-number of your second secretary
nvtini_leave	Defines a two character string to pass to your application
nvtini_speckey	Defines a 15 character string to pass to your application.
nvtini_hrlg	Defines an Integer. Any value from -9999 to 29999 is possible. example: 0816 might mean: open from 08:00 to 16:00

and following...

PMNVT - New Voice Tool - Monitor

Use the pull-down menu CONFIGURATION to modify Startup Parameters. Another fast way to change parameters, especially during a remote session, is to use NVTINI.EXE. At each call of nvtini:exe you can set a parameter to a value for several lines.

Example:

NVTINI	shows help
NVTINI -l	shows all parameters
NVTINI 1	shows all the values of the line 1
NVTINI 3 32	shows all the values of the line 3 to 32
NVTINI 1 progname demo	sets progname of line 1 to demo.nve
NVTINI 9 18 progname demo	sets progname of line 9 to 18 to demo.nve

The printout of the line description for line may look like the following:

```
line <01> of New Voice Tool line descriptor file NVTINI.DBF

lg1      c:\dtvox\          lg2      c:\frvox\
dbv1     c:\vmvox\         dbv2     .
db1      c:\nv\           db2      .
aab1     VM               aab2     .
ts1      112              ts2      .
din      00002            tt        199
amt      A                leave     D
rings    01               speckey  .
us3      .                us4      .
hrlg     000000           recqu    3
maxrec   0080             telldir  c:\nv
progname nvms1600         progdir  c:\my\pl
statname stat.log        statdir  c:\my\pl
nvtini_s1 s1             nvtini_s2 s2
nvtini_s3 s3             nvtini_s4 s4
i1       00001            i2       00002
i3       00003            i4       00004
```

The elements lg1, lg2,.. i4 are described on the next page

Line Startup Parameters -cont III

Additional parameter description. Usefull with **New Voice Mail Power Functions**
 These parameters are not any more used in this form anymore (example only).

nvtini_din Defines the characteristics of the New Voice Mail port
nvtini_tt Defines additional parameters in conjunction with **nvtini_din**

nvtini_din= 0: Standarddefinition of the New Voice Mail port to distinguish between direct calls and diverted calls and internal and external calls.

nvtini_tt --

nvtini_din= 1: A menu will be played to the caller with the names of the persons to be reached by entering a number given with the name. The filename of the table and the voxfile is stored in **nvtini_tt**.

nvtini_tt <file-table-name>
 default file-table-name is **ttable**

ttable.lst The structure of the table is the following:

```
11:111:Secretary
12:112:Mister Miller
14:114:Miss Thomson
15:121:Secretary 2
16:122:Technical Support
17:131:Sales
18:141:Marketing
19:161:Administration
```

ttable.vox Speaks the list-information to the caller.

nvtini_din= 1: Asks the mailbox number to leave a message. **nvtini_tt** must be empty.

nvtini_tt ""
 The voice-mail-interface should not be activated.
 If **nvtini.speckey= "M*"** and the * - key is pressed instead of a mailbox-number, the caller returns to the mainmenu

nvtini_din= 2: The caller will be asked the password of the mailbox indicated in **nvtini_tt** to listen to the recorded messages.

nvtini_tt <mailbox-number>

nvtini_din= 6: The caller will be transferred to the phone number indicated in **nvtini_tt**.

nvtini_tt <phone-number>

nvtini_din= 7: The caller will enter an internal client-solution indicated in **nvtini_tt**.

nvtini_tt <application_name>

Line Startup Parameters -cont IV

Parameter: Value settings for the use of NVMS1600: New Voice Mail Power Functions

field-name default value for NVMS1600 < line 1 to 40 > your configuration

nvtini_ig1	c:\dtvox\	:
nvtini_ig2	c:\frvox\	:
nvtini_dbv1	c:\vmvox	:
nvtini_dbv2		:
nvtini_db1	c:\db1	:
nvtini_db2		:
nvtini_aab1	VM	:
nvtini_aab2		:
nvtini_amt	A	:
nvtini_rings	2	:
nvtini_maxrec	90	:
nvtini_recqu	2	:
nvtini_progname	nvms1600	:
nvtini_progdir	c:\nvt	:
nvtini_statname	stat.log	:
nvtini_statprg	c:\nvt	:
nvtini_maindir	c:\nvt	:
nvtini_teldir	c:\nvt	:

Configuration file lines 1 to 5

Parameter: Note the value settings of your configuration for each line independently

field-name	line1	line2	line3	line4	line5
nvtini_lg1					
nvtini_lg2					
nvtini_dbv1					
nvtini_dbv2					
nvtini_db1					
nvtini_db2					
nvtini_aab1					
nvtini_aab2					
nvtini_amt					
nvtini_rings					
nvtini_maxrec					
nvtini_recqu					
nvtini_progname					
nvtini_progdir					
nvtini_statname					
nvtini_statprg					
nvtini_maindir					
nvtini_teldir					
nvtini_i1					
nvtini_i2					
nvtini_i3					
nvtini_i4					
nvtini_s1					
nvtini_s2					
nvtini_s3					
nvtini_s4					

Configuration file lines 6 to 10

Parameter: Note the value settings of your configuration for each line independently

field-name	line6	line7	line8	line9	line10
nvtini_lg1					
nvtini_lg2					
nvtini_dbv1					
nvtini_dbv2					
nvtini_db1					
nvtini_db2					
nvtini_aab1					
nvtini_aab2					
nvtini_amt					
nvtini_rings					
nvtini_maxrec					
nvtini_recqu					
nvtini_progname					
nvtini_prokdir					
nvtini_statname					
nvtini_statdir					
nvtini_maindir					
nvtini_teldir					
nvtini_i1					
nvtini_i2					
nvtini_i3					
nvtini_i4					
nvtini_s1					
nvtini_s2					
nvtini_s3					
nvtini_s4					

Configuration file lines 11 to 15

Parameter: Note the value settings of your configuration for each line independently

field-name	line11	line12	line13	line14	line15
nvtini_lg1					
nvtini_lg2					
nvtini_dbv1					
nvtini_dbv2					
nvtini_db1					
nvtini_db2					
nvtini_aab1					
nvtini_aab2					
nvtini_amt					
nvtini_rings					
nvtini_maxrec					
nvtini_recqu					
nvtini_progname					
nvtini_prokdir					
nvtini_statname					
nvtini_statdir					
nvtini_maindir					
nvtini_teldir					
nvtini_i1					
nvtini_i2					
nvtini_i3					
nvtini_i4					
nvtini_s1					
nvtini_s2					
nvtini_s3					
nvtini_s4					

Configuration file lines 16 to 20

Parameter: Note the value settings of your configuration for each line independently

field-name	line16	line17	line18	line19	line20
nvtini_lg1					
nvtini_lg2					
nvtini_dbv1					
nvtini_dbv2					
nvtini_db1					
nvtini_db2					
nvtini_aab1					
nvtini_aab2					
nvtini_amt					
nvtini_rings					
nvtini_maxrec					
nvtini_recqu					
nvtini_progname					
nvtini_prokdir					
nvtini_statname					
nvtini_statdir					
nvtini_maindir					
nvtini_teldir					
nvtini_i1					
nvtini_i2					
nvtini_i3					
nvtini_i4					
nvtini_s1					
nvtini_s2					
nvtini_s3					
nvtini_s4					

Configuration file lines 21 to 25

Parameter: Note the value settings of your configuration for each line independently

field-name	line21	line22	line23	line24	line25
nvtini_lg1					
nvtini_lg2					
nvtini_dbv1					
nvtini_dbv2					
nvtini_db1					
nvtini_db2					
nvtini_aab1					
nvtini_aab2					
nvtini_amt					
nvtini_rings					
nvtini_maxrec					
nvtini_recqu					
nvtini_progname					
nvtini_prokdir					
nvtini_statname					
nvtini_statdir					
nvtini_maindir					
nvtini_teldir					
nvtini_i1					
nvtini_i2					
nvtini_i3					
nvtini_i4					
nvtini_s1					
nvtini_s2					
nvtini_s3					
nvtini_s4					

Configuration file lines 26 to 20

Parameter: Note the value settings of your configuration for each line independently

field-name	line26	line27	line28	line29	line30
nvtini_lg1					
nvtini_lg2					
nvtini_dbv1					
nvtini_dbv2					
nvtini_db1					
nvtini_db2					
nvtini_aab1					
nvtini_aab2					
nvtini_amt					
nvtini_rings					
nvtini_maxrec					
nvtini_recqu					
nvtini_progname					
nvtini_prokdir					
nvtini_statname					
nvtini_statdir					
nvtini_maindir					
nvtini_teldir					
nvtini_i1					
nvtini_i2					
nvtini_i3					
nvtini_i4					
nvtini_s1					
nvtini_s2					
nvtini_s3					
nvtini_s4					

Additional variables - for internal use only

The New Voice Tool Initialization file proposes additional parameters to pass. They are strictly for internal use

Additional parameters:

nvtini_autorun	internal use, -9 stops the actual line (flag in nvtbatch.dbf)
	internal use, -9 disables job polling (flag in nvtpost.dbf)
nvtini_checktime	internal use,
nvtini_debugger	internal use,
nvtini_devloop	internal use,
nvtini_deb_ident	internal use,
nvtini_firsttime	internal use,
nvtini_viewcode	internal use,
nvtini_win	internal use,

nvtini_f1	Flag 1
nvtini_f2	Flag 2
nvtini_f3	Flag 3
nvtini_f4	Flag 4
nvtini_f5	Flag 5
nvtini_f6	Flag 6
nvtini_f7	Flag 7
nvtini_f8	Flag 8
nvtini_f9	Flag 9

List of all variables - 1. 1. 1998

Please note, that variables can be appended at anytime and therefore an adapted file _init.nve is needed. Please contact your New Voice Tool representative if you have any questions.

With every compilation of any new voice tool program all variables will be listed and added to the program. This helps you, to keep all variables in your mind.

NVT - Compiler demo.nvt

```
int st           // st= play
int Hangup       // caller onhook, while option -cont
int Fastbusy     // true, if fast-busy-tone detected
int Loopdrop     // true, if loop-drop detected
int Notfound     // true, if a voxfile could not be found
int Timeout      // true, if getkey did not receive the expected keys
str key          // getkey key 1 -time 30
int dbok         // dbase-function returncode
int dbint
str dbstr
str nvtini_s1    // one of 4 strings for statistic file
str nvtini_s2
str nvtini_s3
str nvtini_s4
```

int nvtini_linenbr
str nvtini_lg1
str nvtini_lg2
str nvtini_dbv1
str nvtini_dbv2
str nvtini_db1
str nvtini_db2
str nvtini_aab1
str nvtini_aab2
str nvtini_ts1
str nvtini_ts2
int nvtini_din
str nvtini_tt
str nvtini_amt
str nvtini_leave
int nvtini_rings
str nvtini_speckey
str nvtini_us2
str nvtini_us3
str nvtini_us4
int nvtini_hrlg
int nvtini_maxrec
int nvtini_recqu
int nvtini_ui4
str nvtini_progname
str nvtini_prokdir
str nvtini_statname
str nvtini_statdir
str nvtini_maindir
int nvtini_i1
int nvtini_i2
int nvtini_i3
int nvtini_i4
int nvtini_dbflags
int nvtini_ring_sim
int nvtini_stocode
int nvtini_autorun
int nvtini_checktime
int nvtini_debugger
int nvtini_devloop
int nvtini_deb_ident
int nvtini_firsttime
int nvtini_viewcode
int nvtini_win
int nvtini_f1
int nvtini_f2
int nvtini_f3
int nvtini_f4

int nvtini_f5
int nvtini_f6
int nvtini_f7
int nvtini_f8
int nvtini_f9
str nvtini_sn1
str nvtini_sn2
str nvtini_sn3
str nvt_hseconds
str nvt_seconds
str nvt_minutes
str nvt_hours
str nvt_wday
str nvt_yday
str nvt_day
str nvt_month
str nvt_year
str nvt_time
str nvt_date
str nvt_ext_phone
str nvt_int_phone
int nvt_phone_type
int nvtini_linegrp
str nvt_own_phone
str nvt_call_id
str nvt_call_name
str nvt_call_ref
str nvt_call_prev
str nvt_call_client
str nvt_week
str key1
str key2
str key3
str key4
str key5
str key6
str key7
str key8
str key9
str nvtini_s5
str nvtini_s6
str nvtini_s7
str nvtini_s8
str nvtini_s9
str nvtini_s10
str nvtini_s11
str nvtini_s12
str nvtini_s13

str nvtini_s14
str nvtini_s15
str nvtini_s16
str nvtini_s17
str nvtini_s18
str nvtini_s19
str nvtini_s20
int nvtini_i5
int nvtini_i6
int nvtini_i7
int nvtini_i8
int nvtini_i9
int nvtini_i10
int nvtini_i11
int nvtini_i12
int nvtini_i13
int nvtini_i14
int nvtini_i15
int nvtini_i16
int nvtini_i17
int nvtini_i18
int nvtini_i19
int nvtini_i20
str nvt_fname1
str nvt_fname2
str nvt_fname3
str nvt_fname4
str nvt_fname5
str nvt_fname6
str nvt_fname7
str nvt_fname8
str nvt_fname9
str nvt_fname10
str nvtini_lg3
str nvtini_lg4
str nvtini_lg5
str nvtini_lg6
str nvtini_lg7
str nvtini_lg8
str nvtini_lg9
str nvtini_lg10
str nvt_net_path
str nvt_data_path
str nvt_vox_path
int nvt_duration
int nvt_debug
str nvt_ip_addr1
str nvt_ip_addr2

str nvt_ip_addr3
str nvt_ip_addr4
str nvt_ip_addr5
str nvt_port_addr1
str nvt_port_addr2
str nvt_port_addr3
str nvt_port_addr4
str nvt_port_addr5
int nvt_i1_x
int nvt_i2_x
int nvt_i3_x
int nvt_i4_x
int nvt_i5_x
int nvt_i6_x
int nvt_i7_x
int nvt_i8_x
int nvt_i9_x
str nvt_s1_x
str nvt_s2_x
str nvt_s3_x
str nvt_s4_x
str nvt_s5_x
str nvt_s6_x
str nvt_s7_x
str nvt_s8_x
str nvt_s9_x
str nvt_anc_path
str nvt_anc_line
str nvt_procname1
str nvt_procname2
str nvt_procname3
str nvt_procname4
str nvt_procname5
str nvt_procname6
int nvt_progstack
str nvt_key_code

New Voice Initialization

Extended Configuration Flags Additional predefined Variables

Till Version 3.0 of New Voice Tool, predefined variables were limited to the fields in the database nvtini.dbf. Refer to the chapter I to learn more about these variables and how to set default and initial values.

New Variables and Configuration-Flags have been added to increase the number of parameters to be passed from one module to the other and to provide more comprehensive initialization names for new functions, such as sockets and CSTA concepts.

All these Variables and Flags are to be set and modified in the file NVT.INI

To set a default-value to a string variable, apply the following syntax

[strings]

nvtini_s5= New Voice Design Studio, Version 4.0

To set a default-value to an integer variable, apply the following syntax

[integer]

nvtini_i5= 1000000

System-Flags

Transmits nvt_own_phone on out bound calls (ISDN-E1 only) as caller-id.

[D300]

CNXtoPBX= 1

The functions play and rec will detect Hangup after receiving a "b" (b6)

[PBX]

EndOnB6= 1

Define the icon on each line: 1:W95, 2: WNT3.51 3: WNT4.0 -1: no icon

Set ICTYPE= 4 (initial setting only) in all systems with a huge number of calls

[MONITOR]

ICTYPE= 0

This variable defines the number of retrials to open an ascii-file

[file]

opentimeout= 0

NVT.INI / <nvt_year, nvt_yday,...>

This variable contains the actual week-number.

str nvt_week

NVT.INI / Caller Entry <key>

These additional 9 variables can be helpfull in conjunction with the function **getkey**

str key1

str key2

str key3

str key4

str key5

str key6

str key7

str key8

str key9

NVT.INI / Language directories <nvtini_lg1, nvtini_lg2>

These additional 8 variables can be helpfull in conjunction with **play** and **rec**

str nvtini_lg3

str nvtini_lg4

str nvtini_lg5

str nvtini_lg6

str nvtini_lg7

str nvtini_lg8

str nvtini_lg9

str nvtini_lg10

NVT.INI / filenames

These additional 10 variables can be helpfull in conjunction with any function

str nvt_fname1

str nvt_fname2

str nvt_fname3

str nvt_fname4

str nvt_fname5

str nvt_fname6

str nvt_fname7

str nvt_fname8

str nvt_fname9

str nvt_fname10

NVT.INI / strings <nvtini_s1, nvtini_s2, nvtini_s3, nvtini_s4>

These additional 16 variables can be helpfull in conjunction with all functions

```
str nvtini_s5  
str nvtini_s6  
str nvtini_s7  
str nvtini_s8  
str nvtini_s9  
str nvtini_s10  
str nvtini_s11  
str nvtini_s12  
str nvtini_s13  
str nvtini_s14  
str nvtini_s15  
str nvtini_s16  
str nvtini_s17  
str nvtini_s18  
str nvtini_s19  
str nvtini_s20
```

NVT.INI / integer <nvtini_i1, nvtini_i2, nvtini_i3, nvtini_i4>

These additional 16 variables can be helpfull in conjunction with all functions

```
int nvtini_i5  
int nvtini_i6  
int nvtini_i7  
int nvtini_i8  
int nvtini_i9  
int nvtini_i10  
int nvtini_i11  
int nvtini_i12  
int nvtini_i13  
int nvtini_i14  
int nvtini_i15  
int nvtini_i16  
int nvtini_i17  
int nvtini_i18  
int nvtini_i19  
int nvtini_i20
```

NVT.INI / Multy-System Concept

These additional 3 variables can be helpfull in conjunction with the Network-Manager

```
str nvt_net_path  
str nvt_data_path  
str nvt_vox_path
```

NVT.INI / Timer

This variable contains the duration of the call, after calling the **get -time** function

```
int nvt_duration
```

NVT.INI / Debug-Flag

This variable defines the debug-level. 0: no information will be displayed.

```
int nvt_debug
```

NVT.INI / IP-Address / Port-Address

These variables can be usefull in conjunction with all socket-functiions

```
str nvt_ip_addr1  
str nvt_ip_addr2  
str nvt_ip_addr3  
str nvt_ip_addr4  
str nvt_ip_addr5  
  
str nvt_port_addr1  
str nvt_port_addr2  
str nvt_port_addr3  
str nvt_port_addr4  
str nvt_port_addr5
```

NVT.INI / CSTA

These variables can be usefull in conjunction with all CSTA-functiions

```
str nvt_call_id  
str nvt_call_name  
str nvt_call_ref  
str nvt_call_prev  
str nvt_call_client
```

NVT.INI / Interprocess Variable content

```
int nvt_i1_x
int nvt_i2_x
int nvt_i3_x
int nvt_i4_x
int nvt_i5_x
int nvt_i6_x
int nvt_i7_x
int nvt_i8_x
int nvt_i9_x
str nvt_s1_x
str nvt_s2_x
str nvt_s3_x
str nvt_s4_x
str nvt_s5_x
str nvt_s6_x
str nvt_s7_x
str nvt_s8_x
str nvt_s9_x
```

These variables store the actual value of variable content of any process in communication. Dataload passes through the get - shared command, reading the shared memory of the indicated line

NVT.INI / Indicating the actual program / program-level

```
str nvt_procname1
str nvt_procname2
str nvt_procname3
str nvt_procname4
str nvt_procname5
str nvt_procname6
```

NVT.INI / Containing the number of tax pulses during an outbound call

```
int nvt_taximpulse
```

NVT.INI / Contains additional information about any function

```
str nvt_st
```

New Voice Post-Job

nve.exe nve.exe run a program without any telephone line

Name:	nve.exe	program [line_nr] [options]
Inputs:	program	the program to launch
Options:	line_nr -s1textinfo -s2textinfo -s3textinfo -s4textinfo -i112345 -i254321 -i31 -i422222	if > 0: initialize from nvtini.dbf textinfo stored in nvtini_s1 textinfo stored in nvtini_s2 textinfo stored in nvtini_s3 textinfo stored in nvtini_s4 integer stored in nvtini_i1 integer stored in nvtini_i2 integer stored in nvtini_i3 integer stored in nvtini_i4

Includes: environment of nvt.exe, pmnvt.exe, nvd.exe

Category: execution New Voice Tool - Swiss

❑ Description

NVE.EXE allows one to run a program without any telephone line. This can be very useful whenever calculations must be done or a process must be suspended during a specific number of minutes or seconds.

❑ Example

```

/*****
    mbx_list.nvt    program lists all messages of the mailbox
                   call    nve.exe    mbx_list    1    -i1xxxx
*****/

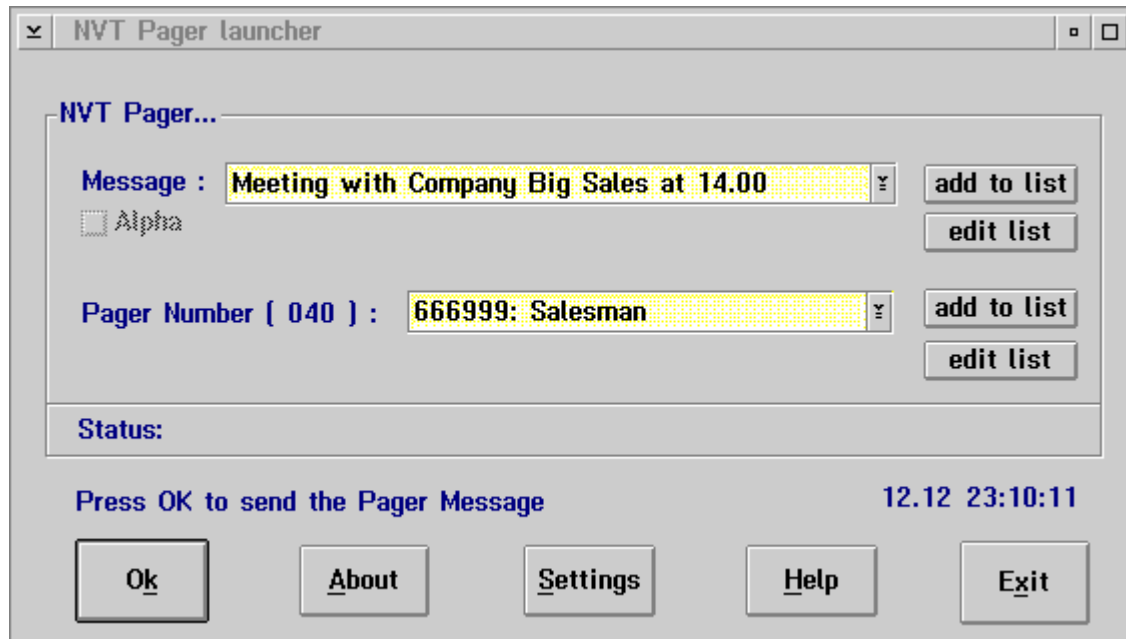
str filename
int i
{
  playmbx    nvtini_i1    -first    filename
  while st> 0
    !"mbx "+ nvtini_i1+ ": msg: "+ filename
    playmbx    nvtini_i1    -next    filename
  wend
  !""
  !"number of messages found: "+ i
}

```

}

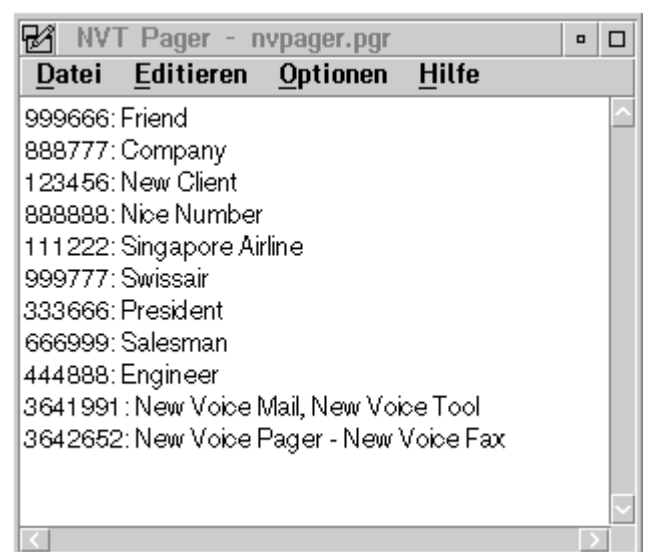
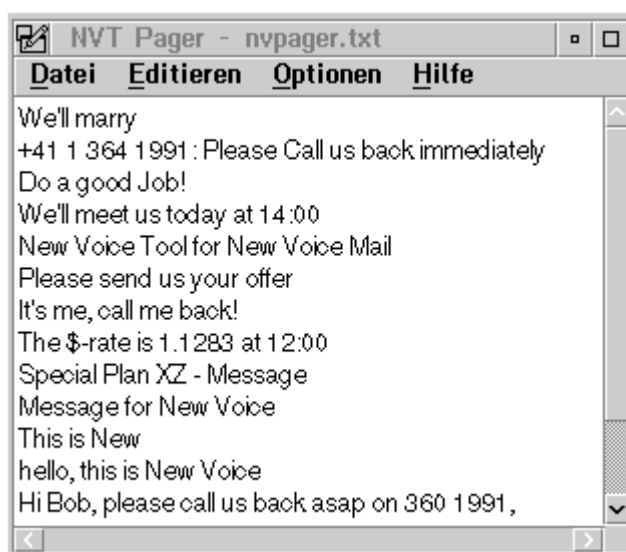
New Voice Pager / SMS

Graphic User Interface



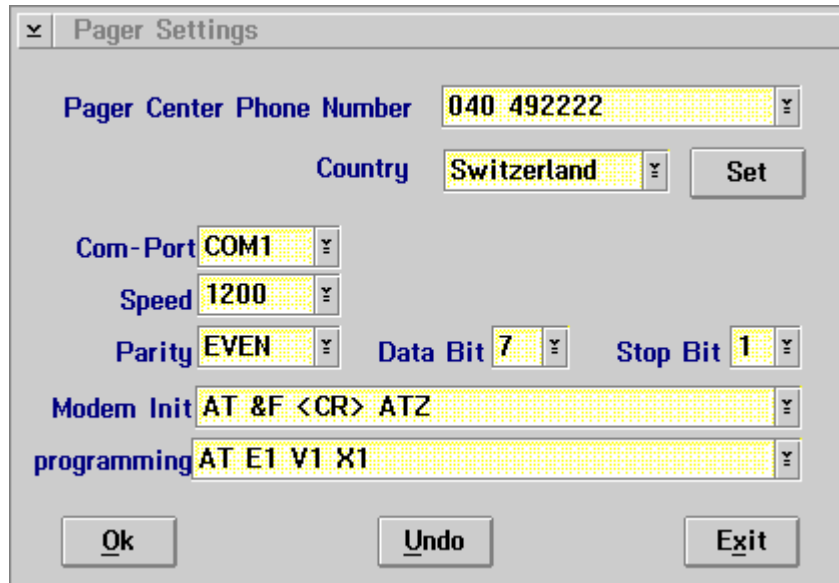
New Voice Pager is an add-on to New Voice Mail - New Voice Tool. A program, especially designed for alpha-numeric messages sent to Pagers. This program needs a modem connected to a com-port of your PC.

This window shows the graphic user interface to send a message from within a terminal. Every message and every pager-number can be added to a list for a fast access in the future.



Note: To send a message from any NVT-Program, simply use the following command
[nvt\nvpager]NVTPAGER.EXE sendpager , as described in the annex.

Settings

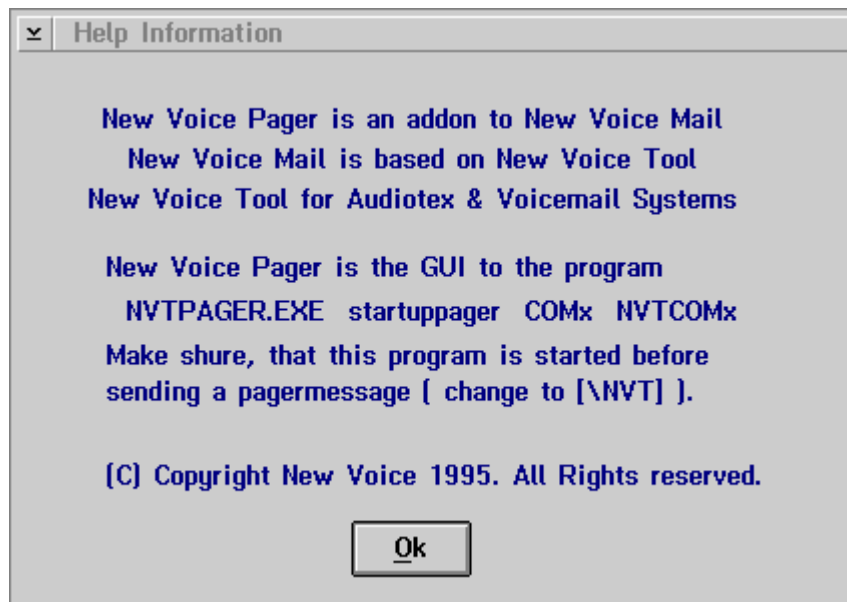


The 'Pager Settings' dialog box contains the following fields and controls:

- Pager Center Phone Number:** Text field with value '040 492222'.
- Country:** Dropdown menu with value 'Switzerland' and a 'Set' button.
- Com-Port:** Dropdown menu with value 'COM1'.
- Speed:** Dropdown menu with value '1200'.
- Parity:** Dropdown menu with value 'EVEN'.
- Data Bit:** Dropdown menu with value '7'.
- Stop Bit:** Dropdown menu with value '1'.
- Modem Init:** Text field with value 'AT &F <CR> ATZ'.
- programming:** Text field with value 'AT E1 V1 X1'.
- Buttons:** 'Ok', 'Undo', and 'Exit'.

The com-port must be selected. Depending the country, use the presets or your specific configuration.

Set-Up



The 'Help Information' dialog box contains the following text:

New Voice Pager is an add-on to New Voice Mail
 New Voice Mail is based on New Voice Tool
 New Voice Tool for Audiotex & Voicemail Systems

New Voice Pager is the GUI to the program
 NVTPAGER.EXE startuppager COMx NVTCOMx
 Make sure, that this program is started before
 sending a pagermessage (change to [NVT]).

[C] Copyright New Voice 1995. All Rights reserved.

Ok

Note: Before sending any message, start [nvt\nvpager]NVPAGER.EXE startuppager comx nvtcomx

nvtpager.exe startuppager

Start Com-Port-Process

Name:	nvtpager.exe startuppager <com-port> <job/data-pipe>	
Inputs:	str com-port	com-connection to modem
	str job/data-pipe	pipe to get messages to send
Options:	none	
Returns:	none	
Includes:	nvtpager.exe sendpager	to post the message
	nvtpager.exe	graphic user interface
Category:	Pager-Commands	New Voice Tool - Swiss

□ Description

This function starts the communication process with the modem to send alpha numeric data to pagers.

To start nvtpager.exe and nvtpager.exe, use the command file NVTPAGER.CMD comx nvtcomx:

```
rem Program to launch nvtpager/nvtpager New Voice Mail-New
Voice Tool
rem nvtpager.cmd <x>      ( x= 1: com1, x= 2: com2 )
start nvtpager startuppager com%1 nvtcom%1
nvtpager
```

To send a message, use the graphic user interface or use the following syntax:

[nvt\nvtpager]nvtpager.exe sendpager comx filename

where

comx is the comport, the modem is connected to
filename is the file containing supplemental information.

The structure of filename is:

<filename>	i.e.	nvp10yyy.pag
<pagercenter>	i.e.	040 492222
<pager-id>	i.e.	666999
<message>	i.e.	Meeting with company Big Sales at 14.00
<modem initial-string>	i.e.	at &f <cr> atz
<modem programming>	i.e.	at e1 v1 x1

Also possible is: nvtpager sendpager comx callcenter pager-id message modem-configuration. See [nvt\nvtpager]nvtpager help for additional information.

New Voice Statistics

Statistics:

The standard statistic file is called **callYYMM.log** and is situated in the "main-dir" directory of your New Voice Tool (nvtini.maindir). The YYMM will be replaced on runtime against the actual year and month, i.e. CALL9406.LOG for june 1994. It is a comma delimited ASCII-file, which allows to treat the data with any calculation-program, such as Excel.

The structure is the following:

DD.MM, hh:mm, ssss, extnumber, intnumber, s1, s2, s3, s4, i1, i2, i3, i4

Where DD.MM is the day and the Month of the year, hh:mm is the time of the day and 'ssss' is the duration of the call in seconds. Extnumber and Intnumber are used for New Voice Mail application only. S1.. S4 and I1.. I4 are the string- and integer-variable nvtini_s1.. nvtini_s4 and nvtini_i1.. nvtini_i4 from the initialisation-file NVTINI. As these 8 variables are global in your system, you can write any relevant information to your .LOG-file by writing them to these variables before going onhook.

In addition to the statistics written into callYYMM.log the contents of i1..i4 and s1.. s4 will be written into the application-specific file indicated on startup in nvtini_statname and nvtini_statdir. This direction can be changed on runtime. Also it is possible to write in any moment into any ascii-file by using the command `fwriteln filename "string"/variable`.

Statistics: cont.

In addition to the file callYYMM.log you can store information to a line-related statistic file. The name of the file is stored in NVTINI.STATNAME and the directory in STATDIR.

The structure is the following:

DD.MM, hh:mm, ssss, s1, s2, s3, s4, i1, i2, i3, i4

Where DD.MM is the day and the Month of the year, hh:mm is the time of the day and 'ssss' is the duration of the call in seconds. S1 to S4 are the initial strings set with the NVTINI command. I1 to I4 are the initial integers set with the NVTINI command.

Attention: Since there can be lots of calls every day, your .LOG-files increases constantly. As big files are difficult to handle, NVT uses CALLYYMM.LOG. That means that every month a new .LOG-file will be created. For January 94 you'll find the file CALL9401.LOG and for July 94 you'll find CALL9407.LOG

For the STATNAME-file, the following rules will be applied:

If the filename contains consecutive capital letters, such as Y M D, they will be exchanged with the actual values of year, month and day.

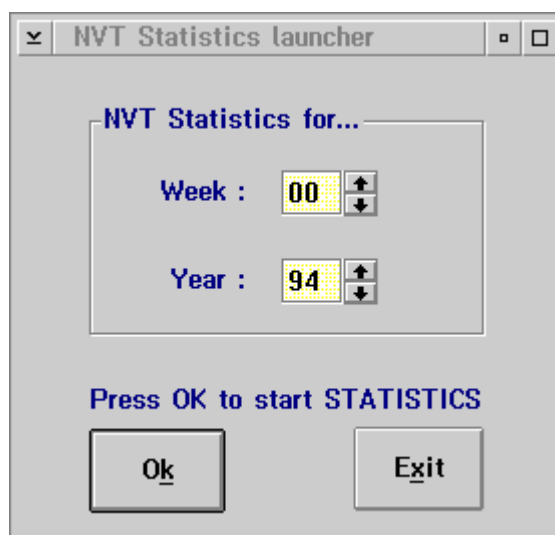
The following keywords are possible: Y, YY, MM, DD, DDD and combinations.

Y :	writes the last significant digit of the year:	4
YY :	writes the two last significant digits of the year:	94
MM :	writes the month in two digits [01.. 12]:	07
DD :	writes the day of month in digits [01.. 31]:	16
DDD:	writes the day of year in digits [001.. 366]:	196

Example: The NVTINI.STATNAME YY_MM_DD.LOG will be transformed to the
following filename: 94_07_16.LOG on July 16th of 1994.

Statistics: cont.

To printout your statistics, use the STATISTIC-menu in the PMNVT program. Select the year and week and start your statistic-job by pressing the **START** button. This will launch the command-file **STATWEEK.CMD** with the parameters %1 year and %2 week.



As it is up to you or your client, to decide, how to proceed after pressing the **START** button, the content of **STATWEEK.CMD** is actually as following:

```
REM STATWEEK.CMD
@ECHO OFF
ECHO The two PARAMS ARE : %1 %2
pause
```

Press enter to close the window. On a client-site, the file could look like this:

```
@ECHO OFF
ECHO The two parameters for STATWEEK.CMD are: %1 %2
nvtcmd.exe statweek \nvt\log\stat%1%2.log
\nvt\log\stat%1%2.prn %2
print \nvt\log\stat%1%2.prn
```

Use the **TYPE** command to add headers and footers to the printed file.

Statistics: cont.

The function library **NVTCMD.EXE** contains several functions to create summaries about the calls. The functions **statweek**, **statdate**, **statappl** and **stathour** open the given log-file and calculate a printable ascii-file with cumulated date/time and duration table.

On OS/2-Command-level [C:\NVT] type: NVTCMD.EXE help to get the exact syntax of the desired command.

The most used function will be **statweek**, as a week allways represents one monday, tuesday.. and one sunday. The command is:

```
NVTCMD statweek <stat-file> <out-file> <field> <week> [cents/minute]
```

STATCONV.NVI STATCONV.NVO:

All functions are using conversion - files to group incoming numbers together and to rename numbers to names.

STATCONV.NVI changes the key-numbers against another key-number in the range 00 - 99. i.e.

36;36;

37;36;

38;38;

39;38;

will add all calls to application 36 & 37 to 36 - and 38 & 39 to 38.

STATCONV.NVO changes the key-numbers against names with 1 to 8 characters.

36;Police;

38;Fire;

will print the names Police and Fire instead of 36 and 38.

STATWEEK.NVI STATWEEK.NVO:

STATCONV.NVI and STATCONV.NVO are used by all functions statweek, statdate, statappl and stathour. Whenever you'd like to work with independent files for each function, use the corresponding files STATWEEK.NVI, STATWEEK.NVO, and STATDATE.NVI, STATDATE.NVO and so on.

Note: These files, if existing, will replace the default-files STATCONV.NVI and .NVO

If you'd like to calculate different statistics using different *.nvi / *.nvo - files, change into different directories before starting the calculations. The *.nvi / *.nvo - files will allways been taken from the actual .\ directory.

Statistics: cont.

The programs **NVTCMD _ STATDATE** and **NVTCMD _ STATAPPL** open the given log-file and calculate a printable ascii-file with cumulated date/time and duration table.

On OS/2-Command-level [C:\NVT] type: NVTCMD HELP to get the exact syntax of the command.

```
nvtcmd statdate <stat-file> <out-file> <field> <month> <startday> <stopday>
```

```
nvtcmd statappl <stat-file> <out-file> <field> <appl> <month> <startday> <stopday>
```

stat-file: complete path and filename of statistic-file (caller.log, stat9407.log,)
 out-file: complete path and filename of out-file containing the calculations
 field: offset, where to find the application-number (1.. 4: S1.. S4, 5.. 8: I1.. I4)
 appl: application-number to scan
 month: month of interest.
 startday: first day of month
 stopday: last day included of month

Hints: Use command-files to start a job scanning for a set of applications.

PRTSTAT.CMD

```
rem This commandfile calculates the statistics for one month
rem and then starts a printout.
rem The application-code is stored in the first field (S1).
```

```
del stattemp.txt
rem      stat-file out-file field mt first last
nvtcmd statdate caller.log stattemp.txt 1 %1 1 31

rem      stat-file out-file field appl mt
nvtcmd statappl caller.log stattemp.txt 1 01 %1 1 31
nvtcmd statappl caller.log stattemp.txt 1 02 %1 1 31
nvtcmd statappl caller.log stattemp.txt 1 03 %1 1 31
nvtcmd statappl caller.log stattemp.txt 1 04 %1 1 31
nvtcmd statappl caller.log stattemp.txt 1 05 %1 1 31

print stattemp.txt
exit
```

Start the commandfile by adding the month: [c:\nvt] nvtcmd 7 (july).

New Voice Helpdesk

Compiler Errors

- 0, ""
- 1, " Use '=' instead of '==' "
- 2, " '=' must be followed by a number",
- 3, " Variable must be followed by '=' "
- 4, " CST, INT, STR and PROC must be followed by a Variable",
- 5, " ';' (or ',') is missing",
- 6, " An expression can not start with this symbol",
- 7, " ')' is missing",
- 8, " The previous factor can not end with this symbol",
- 9, " ',' not needed",
- 10, "Incorrect symbol in statement",
- 11, "This variable is not defined",
- 12, "Assignements to constants are not allowed",
- 13, "The assignement operator is the '='"
- 14, ""
- 15, ""
- 16, ""
- 17, "here is a ';' or '}' expected", 17: see next page also
- 18, ""
- 19, "This statement is followed by an incorrect symbol",
- 20, "A relation is expected here",
- 21, "An expression must not contain a procedure",
- 22, "';' is not needed",
- 23, "WEND (or '}'W') without WHILE",
- 24, "FI without IF", //"FI ohne IF",
- 25, "A variable may not be declared twice",
- 26, ""
- 27, ""
- 28, ""
- 29, "This object type is not expected here",
- 30, "This number is too big",
- 31, "String constant is too big",

How to find Syntax Errors

Independent of how many errors has been found during a compilation, try to correct the first, toplevel error. If for example, a variable has not been declared, every occurrence will cause at least one error!

Regarding the philosophy of compilers, we do not stop the compilation at the first error.

When ever you can not locate an error, reduce your program, until it is so small, that the error does not occur any more. Add the parts you cut off before and compile the program to see, which part causes the problem.

One common error is **error 17**

The following situation will end in the error 17:

```
if x= y
  if y= z
    !" x= y= z"
  fi
else
  !" x not equal y"
fi
```

Simply add `else` to the inner `if fi` so that you have `if else fi`.

These two examples are fine:

```
if x= y
  if y= z
    !" x= y= z"
  fi
fi

if x= y
  if y= z
    !" x= y= z"
  else
    !" y not equal z"
  fi
else
  !"x not equal y"
fi
```

System Errors

- 0, 256, "normal termination",
- 1, 255, "standard error",
- 2, 254, "key-error",
- 3, 253, "error program name",
- 4, 252, "error openchannel, possible key-limit reached",
- 5, 251, "missing linenumber parameter at startup",
- 6, 250, "error initialization with nvtini.dbf",
- 7, 249, "error Voicemail with vm001.dbf",
- 8, 248, "error clear history, controlling the boards",
- 9, 247, "error in group",
- 10, 246, "",
- 11, 245, "",

- , 99, "stop process to restart without errorreport to free the memory",