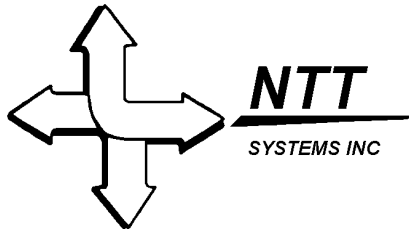


**MECMS**  
**(Mpcsusops)**  
**User Manual**

*NTT Systems Inc.*  
86 Dunblaine Avenue,  
North York, Ontario  
CANADA  
M5M 2S1  
<http://www.ntt.ca>  
**Marc.Grushcow@NTT.ca**

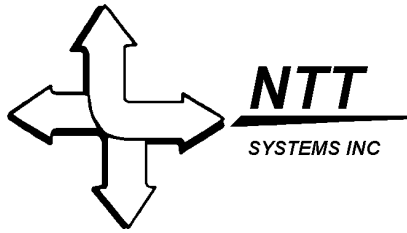




# Table of Contents

<b>1.</b>	<b>Introduction .....</b>	<b>1</b>
<b>2.</b>	<b>Running the Experimenter's Central Console .....</b>	<b>2</b>
<b>3.</b>	<b>Running the Task Dispatchers on the Subjects' Workstations .....</b>	<b>4</b>
<b>4.</b>	<b>Viewing the Dispatchers on the Central Console.....</b>	<b>8</b>
<b>5.</b>	<b>Operating the Central Console .....</b>	<b>10</b>
5.1	<i>The Select Menu .....</i>	10
5.2	<i>The Dispatcher Menu.....</i>	11
5.3	<i>Killing Dispatchers and Tasks .....</i>	14
5.4	<i>The Queue Window .....</i>	15
<b>6.</b>	<b>Scheduled Holds .....</b>	<b>21</b>
<b>7.</b>	<b>Notetaking .....</b>	<b>22</b>
7.1	<i>Adding Notes .....</i>	22
7.2	<i>Viewing Notes.....</i>	24
7.3	<i>Annotating Notes .....</i>	25
7.4	<i>Viewing and Saving the Notes.....</i>	26
<b>8.</b>	<b>Failure and Recovery .....</b>	<b>27</b>
<b>9.</b>	<b>Conclusion.....</b>	<b>28</b>





## 1. Introduction

Every lab engaged in cognitive psychology or similar research has its own methods for creating and running experiments. Ultimately, each one has to deal with experimental design, protocol creation, the logistics of running the experiment, handling failures, data management, and results analysis and interpretation.

[MECMS](#) (Multi-subject Control and Monitoring System) facilitates organizing, running and analysing multi-subject experiments. It is called MpcSusops at [DCIEM](#), reflecting the fact that there it uses the local [Susops](#) (SUStained OPerationS) task battery.

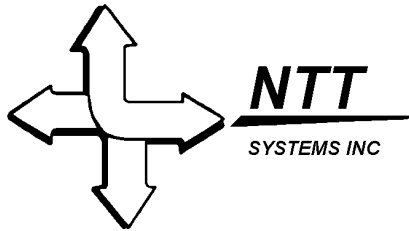
To be more precise, MECMS refers to the core software. It understands things like consoles, workstations, dispatching, and protocols. MpcSusops represents a specific customization of MECMS. It understands what SUSOPS task command lines look like and how parameters (eg., subject IDs) are formatted and used. MECMS technology can be easily customized to accommodate different local task batteries.

The system consists of two types of components: *workstations* and *consoles*. There is one workstation for each subject participating in the experiment and a single console that operates as the experiment controls station. Each workstation is responsible for delivering a predefined series of tasks to the subject using it and reporting progress to the console. The console allows the experimenter to monitor subjects' progress from a central location and provides some intervention mechanisms.

While MECMS can be used for short duration experiments, several features make it especially useful for protocols involving dozens, or even hundreds of tasks. It provides experimenters with a note-taking feature that assists in creating a formatted, legible record of events (eg., unscheduled breaks or medication) and measurements (eg., temperature, blood pressure). In addition, each workstation maintains its own log of task start times and error messages. These can be merged and sorted at the end of an experiment to produce an independently produced record of the protocol that was actually executed.

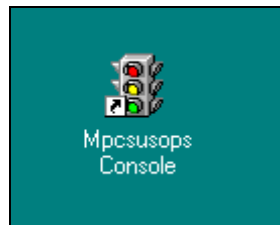
MECMS incorporates a sophisticated, but easy to use, restart and recovery mechanism. If a workstation fails during an experiment, it can be restarted and the console will restore its protocol and state information. This allows the workstation to continue from the task following the point of failure. Trial numbers are properly maintained. If the console fails, each workstation will continue dispatching tasks to its subject. When the console is restarted, it recovers state information automatically from the workstations, including the each workstation's protocol, start times of all tasks, and scheduled modifications to the protocol that were made by the experimenter.

The following tutorial takes you through the basic steps in using the system.



## 2. Running the Experimenter's Central Console

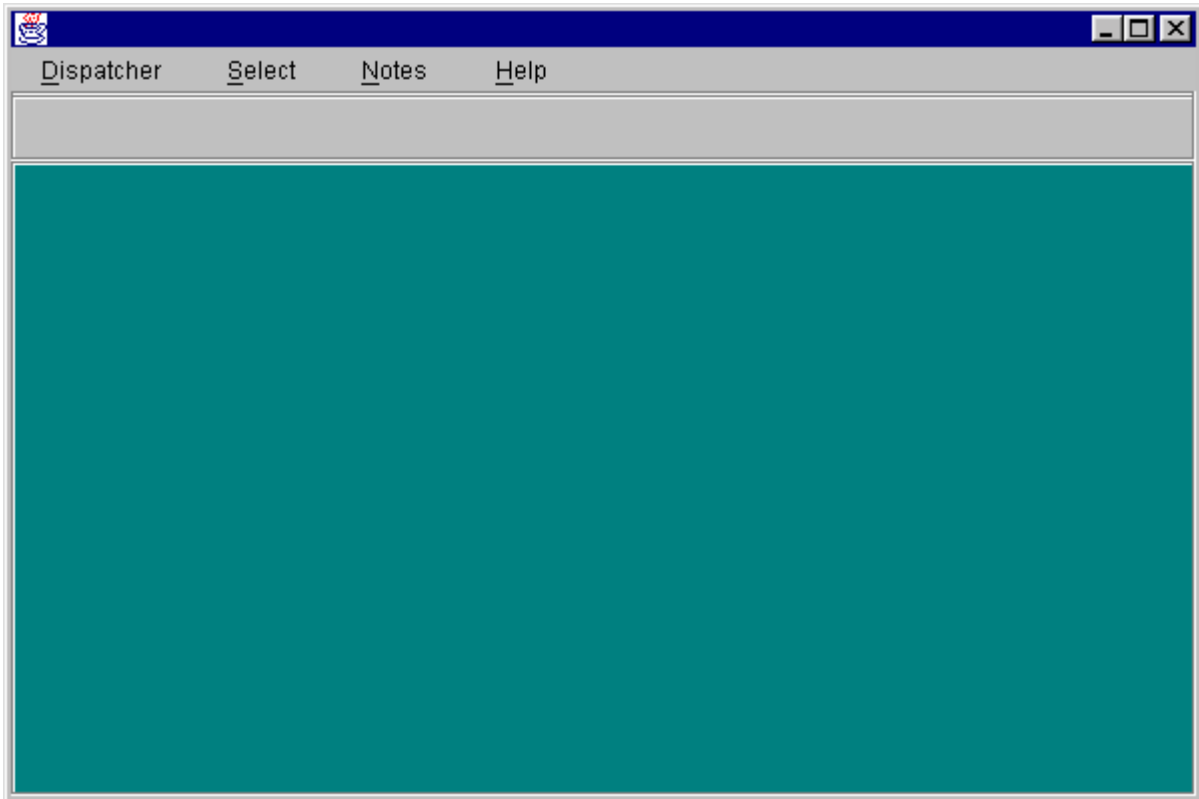
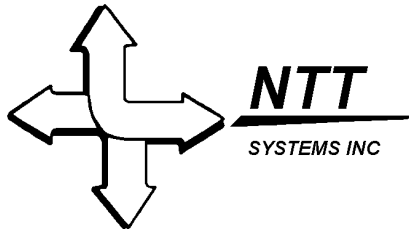
Playing the role of the experimenter, start the system up on the console machine by double-clicking the "Mpcsusops console" icon on the desktop:



When the console starts up, it will ask for the URL of the console machine (in other words, the machine at which you are doing this), as well as the experiment ID. The default URL provided will be correct. The *Experiment ID* is simply a name for the experimenter's log file. It defaults to the last one used. New entries will be appended to the log. This will be the desired action if the console is being restarted after a failure. You should change the name for each new experiment.

A screenshot of a Windows-style dialog box titled "Create Console". The dialog has a blue title bar with a close button. The main area is light gray and contains the text "Please provide the following information:". Below this, there are two input fields. The first is labeled "Console URL:" and contains the text "pcipcontrol2.dciem.dnd.ca:377". The second is labeled "Experiment ID:" and contains the text "expLog". At the bottom center of the dialog is a button labeled "Make Console".

Clicking on the *Make Console* button will bring up the main console window:



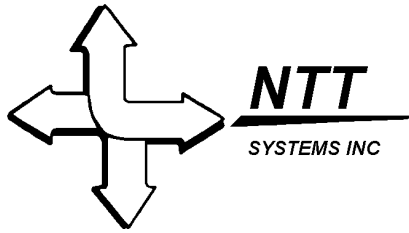
There are four main menus on the console.

The *Dispatcher* menu handles interaction with the dispatcher that runs on each workstation. Menu items facilitate suspending, resuming and terminating dispatching. Recovery features are also accessed from here.

The *Select* menu provides alternate ways to select and deselect workstations.

The *Notes* menu provides access to the note-taking system.

The *Help* menu provides access to the HTML version of this document.

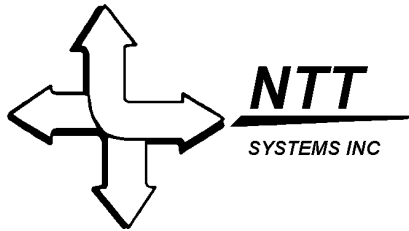


### 3. Running the Task Dispatchers on the Subjects' Workstations

Left to its own devices, the console window will just sit there, blank. To go any further, you need to go start a dispatcher on one of the workstations. Go into one of the subject rooms, and double click the "Mpcsusops Dispatcher" icon on the desktop:

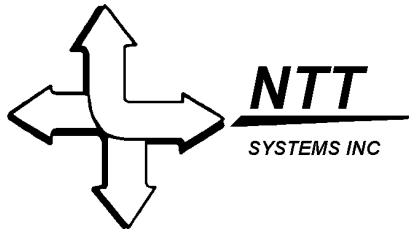


This will bring up an array of parameters for the dispatcher.



Experiment ID	<input type="text" value="ST22"/>	Titan JVM	<input type="text" value="j:\java\bin\java"/>
Subject ID	<input type="text" value="22A"/>	Titan Class Path	<input type="text" value="z:\z:\lib;j:\java\lib\classes"/>
Subject Name	<input type="text" value="Emelie"/>		
Location	<input type="text" value="Alpha"/>		
Protocol File	<input type="text" value="s:\susops\results\ntt\ntt.scr"/>		
Result Directory	<input type="text" value="c:\temp\results"/>		
Exe Directory	<input type="text" value="s:\susops\programs"/>		
Log File	<input type="text" value="c:\temp\results\alpha.log"/>		
Console URL	<input type="text" value="pcipcontrol2:3777/console"/>		
DispatcherURL	<input type="text" value="pctandem5:3777/dispatcher"/>		
TITAN Station ID	<input type="text" value="Alpha"/>		
	<small>eg: Alpha</small>		
TITAN Leader PC Name	<input type="text" value="pctandem4:3666/rt"/>		
	<small>eg: pctandem 3</small>		
<input type="button" value="Create Workstation"/>		<input type="button" value="Restore Workstation"/>	

The fields of the property sheet will be blank if the workstation is being run for the first time in this machine. Otherwise, they will be filled in with values from when the workstation was last created or restored. Any fields including TITAN in the name need only be filled in if the [TITAN](#) Team Decision Making Task will be included in the protocol.



The fields should be filled in as follows:

**Experiment ID** – A string used to identify the workstation on the console display. Also passed to SUSOPS tasks as a parameter for result data identification.

**Subject ID** – A string used to identify the workstation on the console display. Also passed to SUSOPS tasks as a parameter for result data identification.

**Subject Name** – A string used to identify the workstation on the console display and as a memory aid for the experimenter.

**Location** – A string used to identify the workstation on the console display and as a memory aid for the experimenter.

**Protocol File** – The complete path and file name of the protocol file to be used for this experiment.

**Result Directory** – The path to the directory where (SUSOPS) tasks will write their result data. This is passed as a command line parameter to the tasks.

**Exe Directory** – The path to the directory containing the executable files for the task battery.

**Log File** – The complete path and file name that identifies this workstation's log file. Status information is appended to this file.

**Console URL** – The URL of the console as displayed in the console's startup screen.

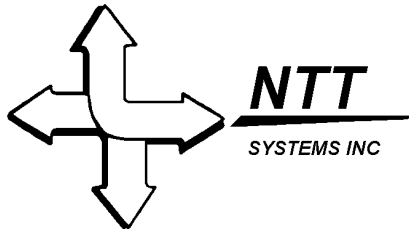
**Dispatcher URL** – The URL of this dispatcher.

**TITAN Station ID** – The TITAN ID for this subject (eg., Alpha, Bravo, Charlie, Leader)

**TITAN Leader PC Name** – The URL of the workstation designated as the Leader for multi-player TITAN runs.

**TITAN JVM** – The full path and file name of the Java Virtual Machine to be used for TITAN runs.

**TITAN Class Path** – The class path identifying the locations of Java class components for TITAN.



Fill in the parameters appropriately and click “Create Workstation”. A window entitled “java” will appear and start up the dispatcher. Wait until the java window says “Workstation ready”, at which point the dispatcher is running in *suspended* mode. It will begin dispatching tasks as soon as the central console gives the *resume* command to all the workstations at once, at which point the experiment begins, with all the scheduled tasks on all the workstations beginning simultaneously.

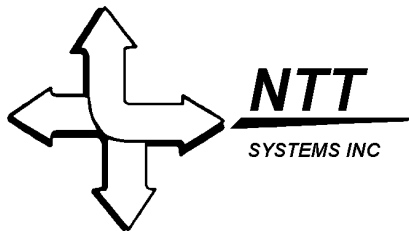
A screenshot of a Windows command prompt window titled "java". The window has a standard Windows interface with a title bar, a menu bar (File, Edit, Format, View, Help), and a toolbar. The command prompt shows the following text:

```
S:\SUSOPS>xcopy z:\lib\ntt\mpcsusops c:\nttjre\lib\ntt\mpcsusops /s /d
0 File(s) copied

S:\SUSOPS>MET.EXE time \\pcipcontrol1 /set /yes
Current time at \\PCIPCONTROL1 is 9-22-1999 11:08A.M.
The command was completed successfully.

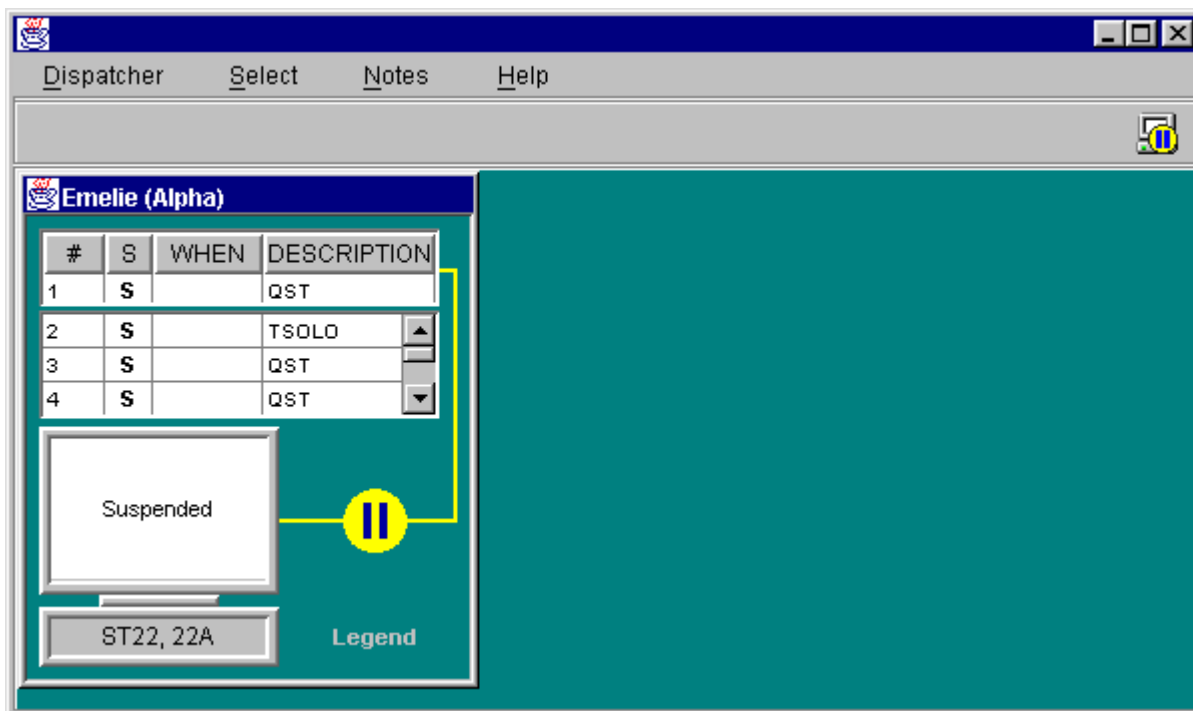
S:\SUSOPS>set CLASSPATH=c:\nttjre\lib;c:\nttjre\jre1.2\lib\rt.jar

S:\SUSOPS>c:\nttjre\jre1.2\bin\java -Djava.security.policy=dispatcher.policy ntt
.mpcsusops.dispatcher.Frame1
Dispatcher bound in the registry.
Workstation ready. Current Dispatchable Task: 0
```



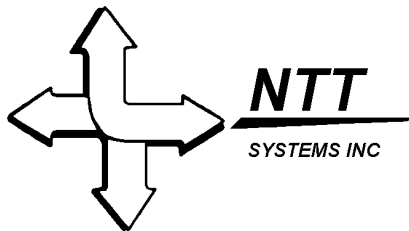
#### 4. Viewing the Dispatchers on the Central Console

The tasks are run entirely on the workstation. The central console simply observes the dispatcher's progress, and allows the experimenter to suspend, resume and cancel tasks. At this point, you can return to the console. You should see that it has automatically detected the presence of a running dispatcher, and is displaying a window describing the status of that workstation. If there is a message box up that says "Creating workstation...", just wait a minute or so for the console to set up its view on the workstation. The following console shows a view on the dispatcher for subject Alpha, whose name is Emelie. Note that this is merely a remote view on the dispatcher, which is still running wholly on the workstation machine in the other room, not on the central console.



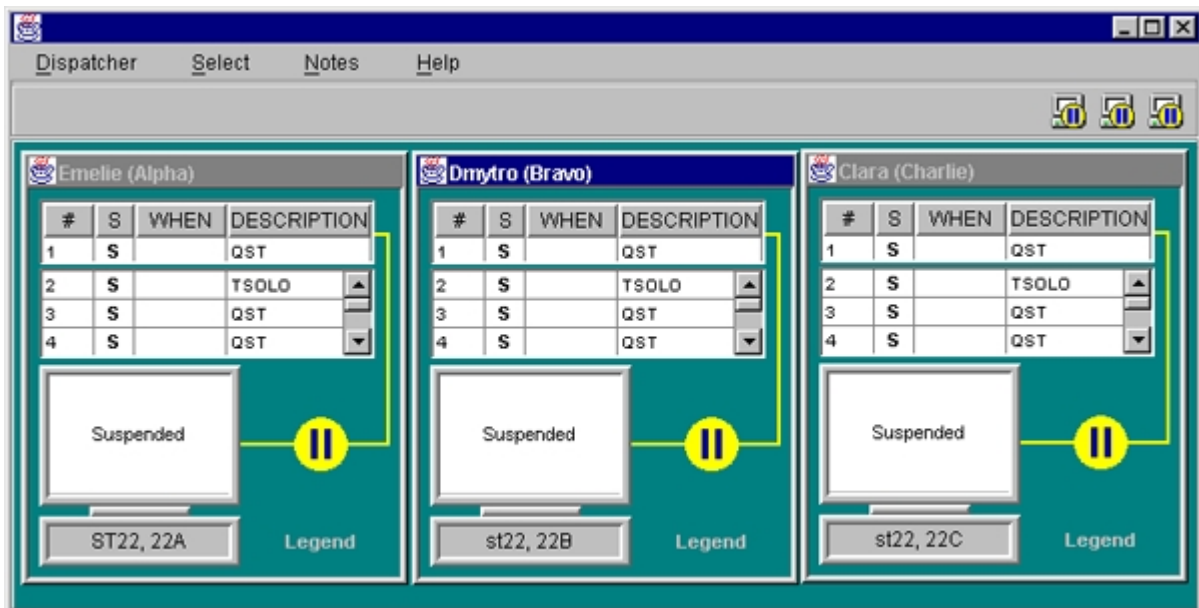
Once you have started dispatchers on all the workstations, the console should have one such window for each workstation. These can be moved around anywhere on the screen, or stacked on top of each other, as shown with the three below.

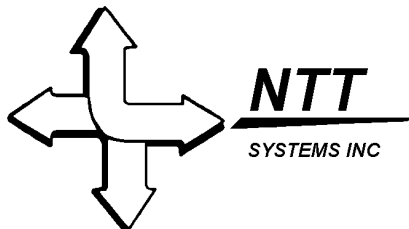
The image inside the window shows a picture of a computer (representing the workstation), connected to the queue of tasks, which is just the list of tasks for that dispatcher, in the order they are to be executed. The tasks are identified by their name, or "description", which is usually an abbreviation of



three or so letters. Also listed for each task is its placement in the queue, current status and, for those tasks that have actually run, the time when they were started . For instance, task #1 in the queue shown above is the “QST” task. It is currently *scheduled*, so its state is “S”, and it has no time, since it has not been run yet.

The numbers at the bottom of the workstation icon (ST22 and 22A in the above example) are the experiment ID and subject ID, respectively. Note that this workstation icon is connected to the task queue by a yellow connector line, with a yellow circle on it, which indicates the current state of the queue. If the dispatcher is suspended, then no tasks are running, and the pause symbol is displayed in the circle, which is the case above. The symbols used here are modelled after the standard symbols used for VCR and tape deck controls.



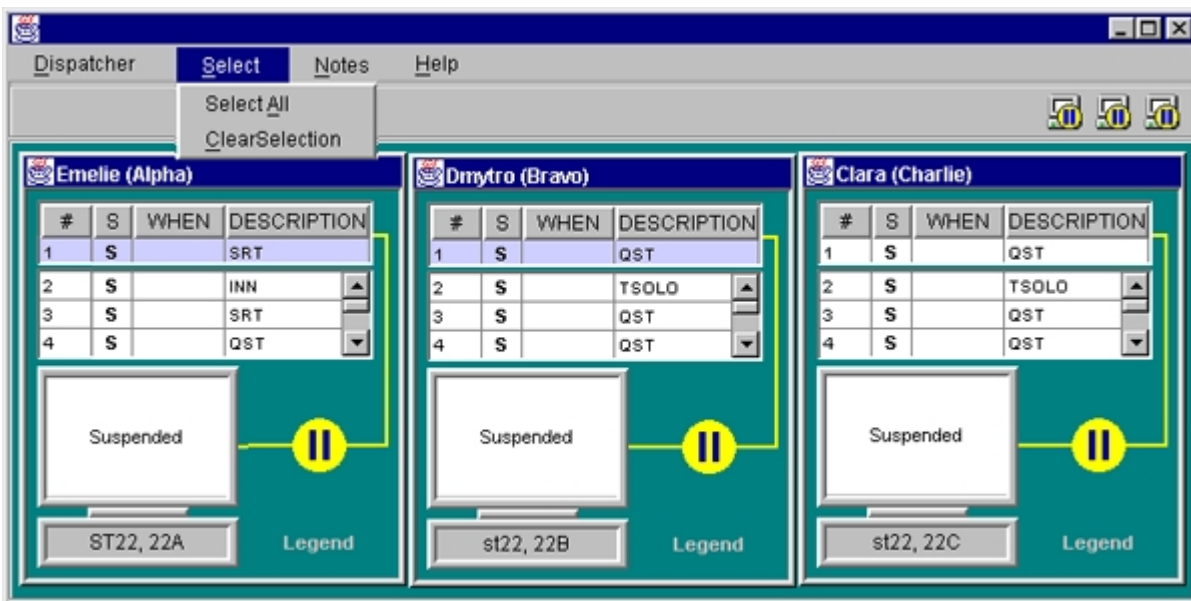


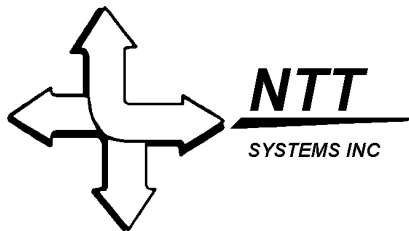
## 5. Operating the Central Console

### 5.1 The Select Menu

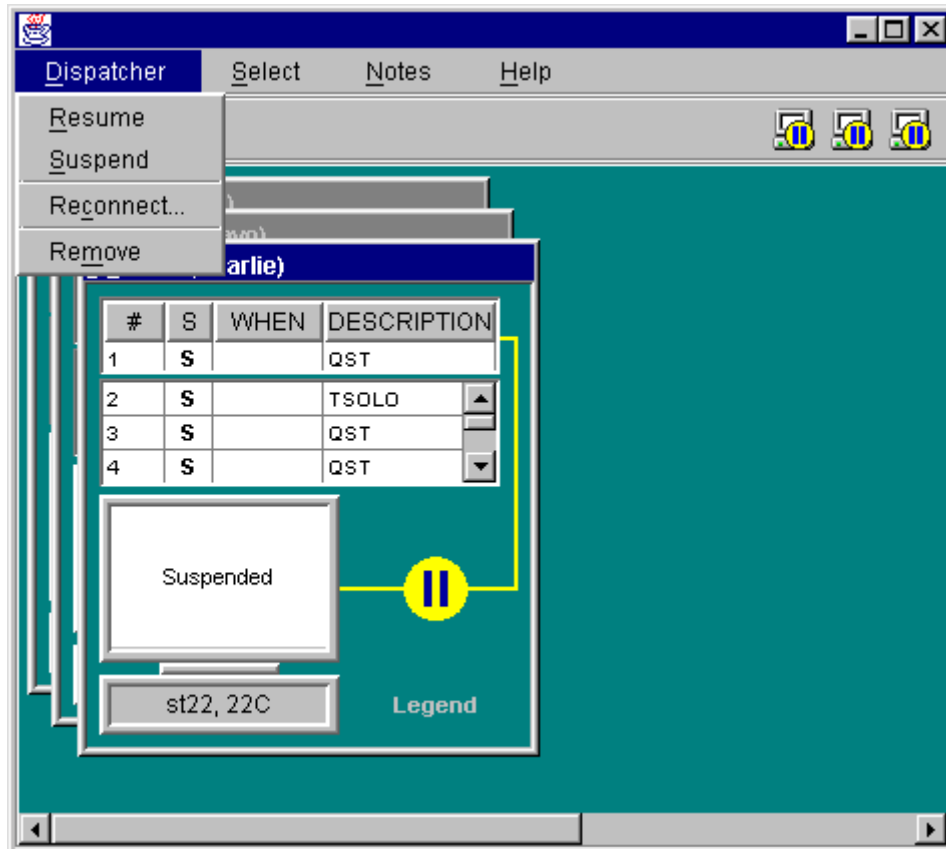
The screen shown above in the last section has one window selected (Bravo). You can tell it is selected because its title bar is highlighted in blue, while the other windows have grey title bars. Note that for each workstation represented in the console, there is a small icon in the top right of the window. A workstation can be selected by clicking on this icon, or by clicking the actual window of the workstation. To select more than one window at a time, hold the SHIFT key down while selecting the windows. To select all the workstations at once, choose “Select All” from the Select menu. To deselect all selected workstations, choose “Clear Selection”.

In the example below, all three workstations are selected, as their title bars are all highlighted. Choosing “Clear Selection” at this point will deselect all the workstations, and the title bars will turn grey. Thus, the workstation windows do not work exactly like regular windows in the Windows operating system, since more than one workstation window can be selected at a time.

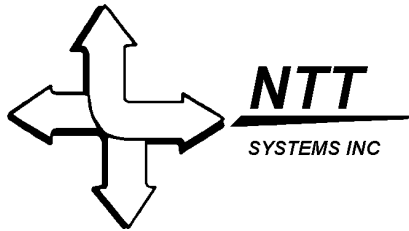




## 5.2 The Dispatcher Menu



The dispatcher menu allows the experimenter to suspend, resume or remove a dispatcher. Since the dispatchers start out suspended, you need to resume them all at the same time at the start of an experiment. Select the workstation or workstations you want to resume, either by clicking on them, on their icons in the upper right corner, or by choosing “Select All” under the Select menu. Then, choose “Resume” under the Dispatcher menu to start the running of all tasks on all selected workstations. Alternatively, you can right-click on the yellow circle for the workstation you want, and choose “Resume” from the pop-up menu, as shown below.



Dispatcher Select Notes Help

Emelie (Alpha) Dmytro (Bravo) Clara (Charlie)

#	S	WHEN	DESCRIPTION
1	S		QST
2	S		TSOLO
3	S		QST
4	S		QST

Suspended

ST22, 22A Legend

#	S	WHEN	DESCRIPTION
1	S		QST
2	S		TSOLO
3	S		QST
4	S		QST

Suspended

st22, 22B Legend

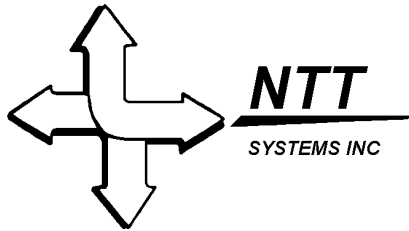
#	S	WHEN	DESCRIPTION
1	S		QST
2	S		TSOLO
3	S		QST
4	S		QST

Suspended

st22, 22C Legend

Resume  
Kill dispatcher  
Kill task

The image shows a screenshot of a software interface for a dispatcher. The window title is "Dispatcher" and it has a menu bar with "Select", "Notes", and "Help". The interface is divided into three main sections, each representing a different dispatcher: "Emelie (Alpha)", "Dmytro (Bravo)", and "Clara (Charlie)". Each section contains a table with columns for "#", "S", "WHEN", and "DESCRIPTION". The "S" column contains the letter "S" for all entries. Below each table is a "Suspended" status indicator and a yellow pause button. At the bottom of each section is a "Legend" button. A context menu is open over the "Dmytro (Bravo)" section, showing options: "Resume", "Kill dispatcher", and "Kill task".



The image of the computer in the little window will now show that the workstation is no longer “suspended”, but is now running tasks. This is indicated by a change in the colour of the little computer’s monitor from white to green, and the display of the three-letter abbreviation for the task that is currently running. As well, the yellow circle now contains the run symbol (a left pointing arrow), instead of the pause symbol. The workstation icon also displays the time the task started, its estimated end time and approximate time left. The example below shows the Bravo workstation running the GTP task, which was task #2 on the queue. That leaves task #3 currently at the top of the queue, awaiting its turn. The start time is shown on the little computer screen as 11:33, along with an estimated end time of 11:37 and approximately two and one half minutes left to go.

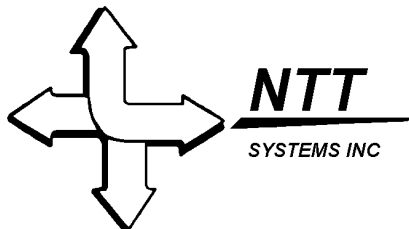
The screenshot shows a workstation interface titled "Dmytro (Bravo)". At the top is a table with the following data:

#	S	WHEN	DESCRIPTION
3	S		QST
2	S		TSOLO
3	S		QST
4	S		QST

Below the table is a green monitor window displaying the following information:

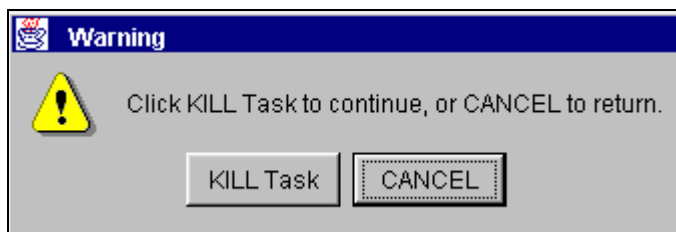
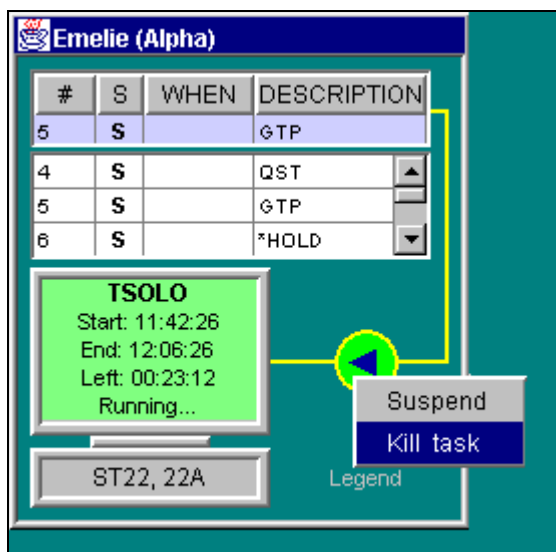
**GTP**  
 Start: 11:33:00  
 End: 11:37:30  
 Left: 00:02:34  
 Running...

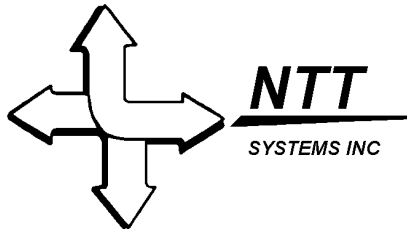
To the right of the monitor is a yellow circle containing a left-pointing arrow. Below the monitor is a grey button labeled "st22, 22B" and the word "Legend".



### 5.3 Killing Dispatchers and Tasks

The pop-up workstation menu also allows us to halt the currently running task by choosing “Kill task”. We can also kill the entire dispatcher for that workstation with “Kill dispatcher”. Killing the dispatcher is, of course, a drastic measure, but even just killing a task is also irreversible. This is not the same as canceling a program that has yet to run. Such tasks can be rescheduled, but killed tasks simply halt their execution and are gone. Therefore, the system will prompt you as to whether you are sure you want to perform the kill for the killing of both tasks and dispatchers.





### 5.4 The Queue Window

So far, our view of the queue only shows tasks waiting in the queue. Double-clicking on the queue in the example above will give a larger view of the queue, as shown below. This is a scrollable list that includes completed tasks and the current running (or suspended) task.

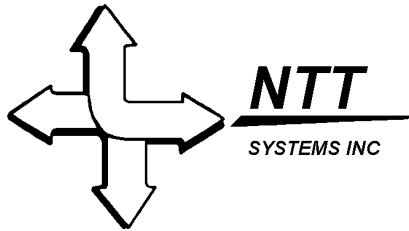
#	S	WHEN	DESCRIPTI...
0	D	12:02 Wed ...	GTP
1	B		QST
2	R	12:02 Wed ...	TSOLO
3	S		QST
4	S		QST
5	S		GTP
6	S		*HOLD
7	C		GTP
8	S		INN
9	S		SRT
10	S		QST
11	S		QST
12	S		INN

Note that the currently running task is highlighted in green and its state is labelled “R” for “running”. When a task is complete, its status is “D” for “Done”, as in task #0 above. Task #7 is marked “C” for “Cancelled.” The dispatcher will skip this task when it gets to it. Task #1 was cancelled and then bypassed, “B” by the dispatcher. Tasks that are cancelled, but not yet bypassed can be returned to the scheduled state. This facility is provided to allow the experimenter to respond to situations where subjects are slow, late returning to their stations, or too fatigued to handle the full work load.

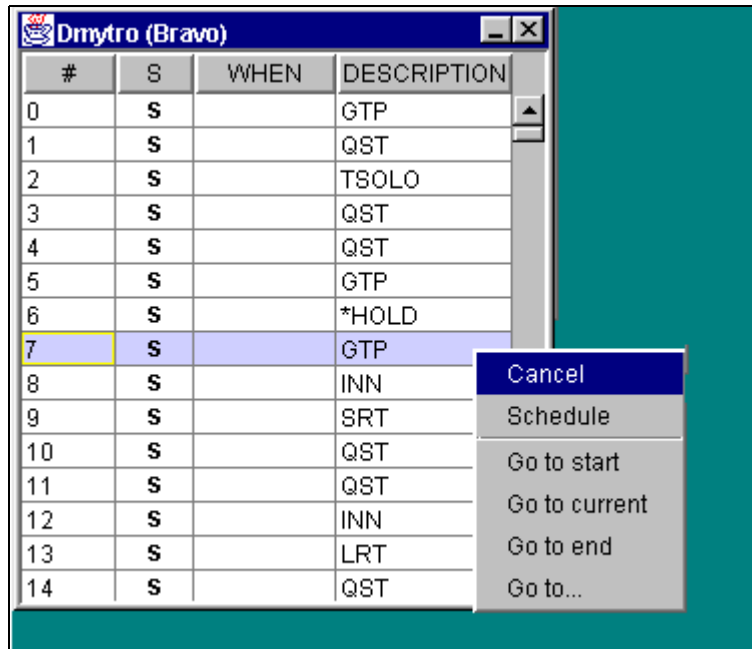
The possible states of a task are:

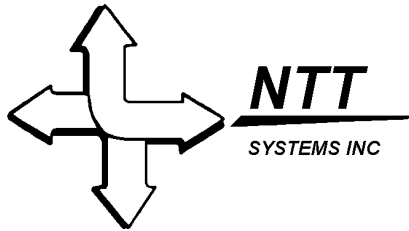
- S: Scheduled
- R: Running
- D: Done
- C: Cancelled
- B: Bypassed
- F: Failed

A task is only marked as Failed if the workstation itself failed when the task was running.



A task can be selected by clicking on it in either the small queue in the workstation window or in the large queue window. Once selected you can cancel it by right clicking on it and selecting “Cancel” from the popup menu. Cancelled tasks can be rescheduled in the same way, but select “Schedule” from the popup.



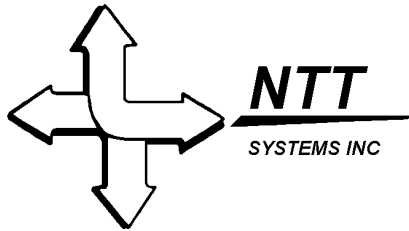


The screenshot shows a software window titled "Dmytro (Bravo)" containing a table with the following data:

#	S	WHEN	DESCRIPTION
0	S		GTP
1	S		QST
2	S		TSOLO
3	S		QST
4	S		QST
5	S		GTP
6	S		*HOLD
7	C		GTP
8	S		INN
9	S		SRT
10	S		QST
11	S		QST
12	S		INN
13	S		LRT
14	S		QST

A context menu is open over row 7, showing the following options: Cancel, Schedule (highlighted), Go to start, Go to current, Go to end, and Go to...

More than one task can be selected in the queue at a time by using shift-click and control-click in the usual way.

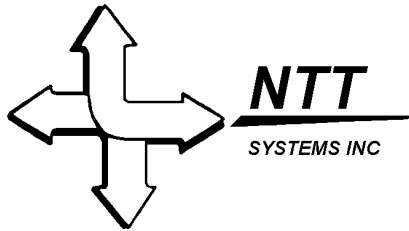


The screenshot shows a window titled "Emelie (Alpha)" containing a table of tasks. The table has four columns: "#", "S", "WHEN", and "DESCRIPTION". The tasks are listed as follows:

#	S	WHEN	DESCRIPTION
49	S		QST
50	S		INN
51	S		SUB
52	S		SUB
53	S		SUB
54	S		SUB
55	S		QST
56	S		QST
57	S		QST
58	S		*HOLD
59	S		QST
60	S		TSOLO
61	S		QST
62	S		QST
63	S		GTP
64	S		QST

A context menu is open over the table, showing the following options: "Cancel", "Schedule", "Go to start", "Go to current", "Go to end", and "Go to...".

Note that in the pop-up menu that, as well as canceling and scheduling, we can also choose to go to the start or end of the queue, to the currently running task, or to a particular task number within the queue. Choosing "Go to..." will bring up a prompt for the task number you wish to go to, and the queue will then display this task at the top of the displayed list, as shown below.



#	S	WHEN	DESCRIPTION
0	S		GTP
1	S		QST
2	S		TSOLO
3	S		QST
4	S		QST
5	S		
6	S		
7	C		
8	S		
9	S		
10	S		
11	S		
12	S		INN

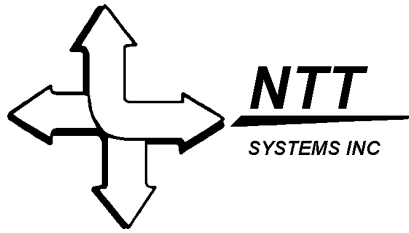
**Go To**

Go to row:

OK Cancel

#	S	WHEN	DESCRIPTION
27	S		TSUB
28	S		QST
29	S		QST
30	S		QST
31	S		QST
32	S		ADD
33	S		QST

These queue operations can also be performed in the smaller version of the queue window. If you are just learning how to use the console, it is recommended that you play around with queue window at this point, and experiment with the different operations.



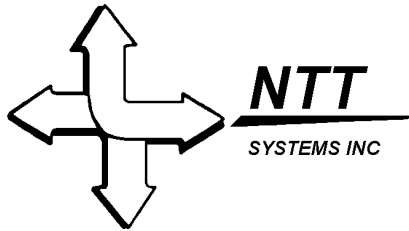
If any task, like the GTP task below goes over its estimated time, the workstation icon will turn yellow instead of green and will indicate a late time, rather than the usual time left.

The screenshot shows a workstation interface for "Emelie (Alpha)". It contains a table with task details and a status window for a task named "GTP".

#	S	WHEN	DESCRIPTION
3	S		QST
2	S		TSOLO
3	S		QST
4	S		QST

**GTP**  
Start: 11:33:00  
End: 11:37:30  
Late2: 00:00:47  
Running...

ST22, 22A      Legend



## 6. Scheduled Holds

Suspending the current task manually from the console is not the way to provide a pre-scheduled pause between tasks. That is done by scheduling pseudo-task called \*HOLD task. Dispatchers treat the \*HOLD task as a command to suspend dispatching. The HOLD has a nominal time associated with it, for example, the time scheduled for a meal break. Note that the VCR control automatically changes to a *pause* icon when a \*HOLD is initiated. Dispatchers must be resumed manually. If the experimenter allows the hold to run longer than the scheduled time, the display will turn yellow and show the over-run time in the usual manner.

The screenshot shows the Emelie (Alpha) control interface. On the left, a task list window displays the following data:

#	S	WHEN	DESCRIPTION
9	S		SRT
8	S		INN
9	S		SRT
10	S		QST

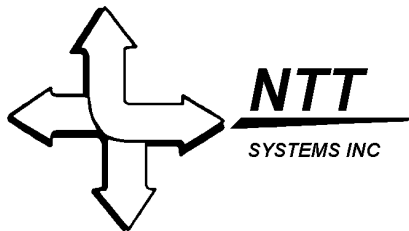
Below the task list, a yellow status window displays the following information:

**\*HOLD**  
 Start: 11:46:08  
 End: 11:46:09  
 Late2: 00:00:03  
 Holding

A yellow pause icon (two vertical bars) is shown next to the status window. Below the status window, a button labeled "ST22, 22A" and a "Legend" label are visible.

On the right, a larger task list window displays the following data:

#	S	WHEN	DESCRIPTION
0	D	11:40 Wed Sep 22 1999	GTP
1	D	11:42 Wed Sep 22 1999	QST
2	D	11:44 Wed Sep 22 1999	TSOLO
3	D	11:44 Wed Sep 22 1999	QST
4	D	11:45 Wed Sep 22 1999	QST
5	D	11:46 Wed Sep 22 1999	GTP
6	R	11:46 Wed Sep 22 1999	*HOLD
7	S		GTP
8	S		INN



## 7. Notetaking

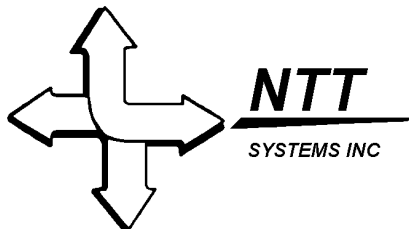
The note-taking system allows you to enter information about the experiment while it is running. It offers a number of advantages over notebooks, not the least of which are increased consistency and legibility.

### 7.1 Adding Notes

A new note is started by selecting “Add” from the notes menu. It presents a form like the following:

A screenshot of a Windows-style dialog box titled "Adding a new note". The dialog has a light blue background. At the top left, there is a "Re:" label followed by a dropdown menu. Below it, a "Date:" label is partially visible. To the right of the "Re:" dropdown is a "Title:" label followed by a text input field. Below the "Title:" field is an "Author:" label followed by another dropdown menu. A large, empty text area occupies the center of the dialog. At the bottom, there are two buttons: "Save" on the left and "Cancel" on the right. A dropdown menu is open below the "Re:" field, showing two options: "software" and "general", with "general" highlighted in blue.

The “Re:” field is used to identify a particular type of issue. Examples might be things like: software, general, medication, meals, etc. Each time you enter new text in this field, it is remembered and offered as a drop down choice next time. The “Title” field should be used to identify the specific event, for example “Gave Fred Tylenol.” The “Author” field operates like “Re:” in that it remembers entered values and offers them as options. The “Date” field is filled in automatically based on information available when you selected the “Add” menu item. To complete the note simply fill in the fields and note body as in the following figure.

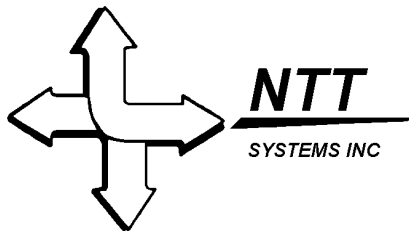


**Adding a new note**

Re:  Title:

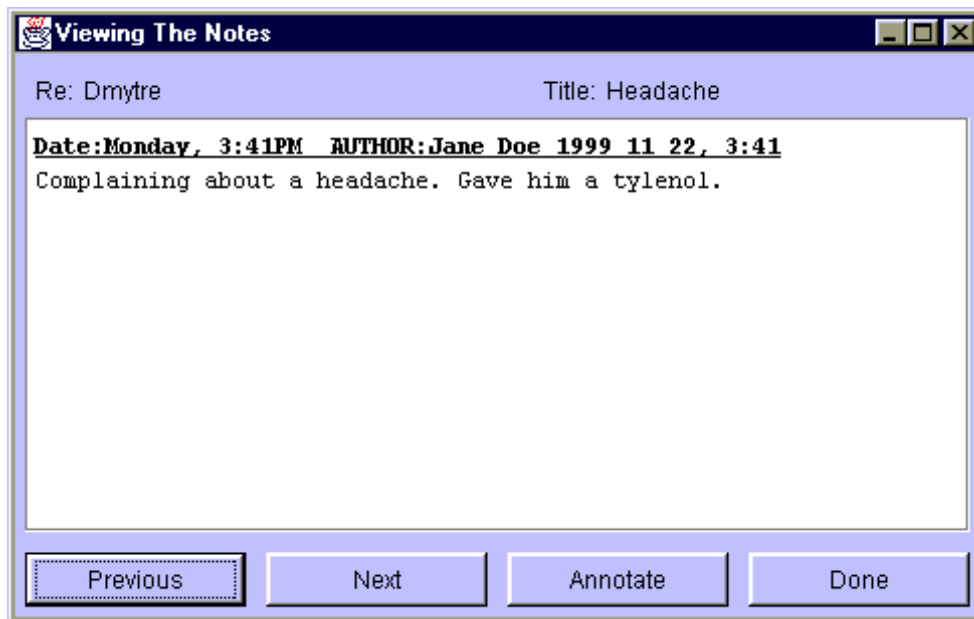
Date: Monday, 3:41PM Author:

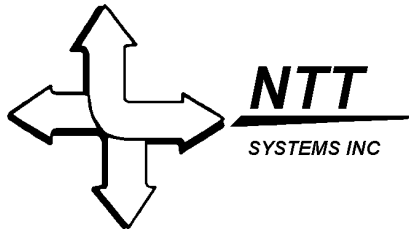
Click on the "Save" button to add it to the log, or the "Cancel" button to discard it.



## 7.2 Viewing Notes

The entire log can be viewed note by note, by selecting Notes->View from the menu. The most recent note is displayed. You can move back and forth in the log by clicking the “Previous” and “Next” buttons. Close the display by clicking “Done”.





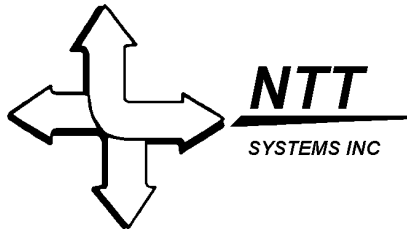
### 7.3 Annotating Notes

You can not change the text of a note once it has been saved, but you can add annotations to it. This is done by first viewing the desired note and then clicking on the “Annotate” button. The form will be changed as shown below. Enter the select/enter the author name and add the text of the annotation.

A screenshot of a Windows-style dialog box titled "Viewing The Notes". The dialog has a light blue background. At the top, it displays "Re: Dmytre" and "Title: Headache". Below this is a text area containing the text: "Date: Monday, 3:41PM AUTHOR: Jane Doe 1999 11 22, 3:41" followed by "Complaining about a headache. Gave him a tylenol." Below the text area, there is a "Date:" field with "Monday, 3:48PM" and an "Author:" dropdown menu currently showing "Jane Doe". A second text area contains the text "Actually, it was two tylenol." At the bottom of the dialog are two buttons: "Save" and "Cancel".

Thereafter, the note will be displayed as follows:

A screenshot of the "Viewing The Notes" dialog box after the annotation process. The layout is similar to the previous screenshot, but the "Date:" and "Author:" fields are no longer present. The text area now displays two entries: "Date: Monday, 3:41PM AUTHOR: Jane Doe 1999 11 22, 3:41" with "Complaining about a headache. Gave him a tylenol." below it, and "Date: Monday, 4:30PM AUTHOR: Jane Doe 1999 11 22, 4:30" with "Actually it was two." below it. At the bottom of the dialog are four buttons: "Previous", "Next", "Annotate", and "Done".



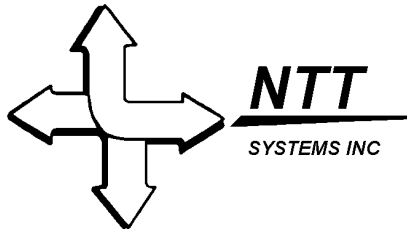
## 7.4 Viewing and Saving the Notes

The internal note file is saved after each modification so that it is protected from system failures. This format can not be printed or modified. The entire log can be viewed at any time by selecting Notes->Preview from the menu. You will be presented with a display like this...

Date	Time	Author	Reference	Title
Monday Oct 18, 1999	7:51P M	Andrea	electrodes	new procedure
We used celodion and gauze on top of the water soluble (Aqua bond) electrodes. It seems to have worked very well and took approximately 90 min.				
Tuesday Oct 19, 1999	7:55A M	Heather & meg	bedtime/reveille	lights out Monday
Ss went to bed at 23:08				
Tuesday Oct 19, 1999	9:33A M	Joe	software	charlie problemo
Charlie's TITAN got hung up upon ending 2nd team run Error message was: "cannot access remote file logger - communication failed - click button to end task"  System did not respond to "exit click" Used new susops feature: right click on green circle button, did a "suspend" and right click again "kill task" then resumed. Seems ok for now. Charlie was about 5 minutes behind the rest. Regards, JVB.				
Wednesday Oct 20, 1999	5:53A M	Heather	general	know about nap
The subjects know about the nap. They probably found out from friends at their unit who participated in the study.				
Wednesday Oct 20, 1999	7:04A M	susan	software	End of ATC - a.m.
ATC - Leader's computer screen did not show "Continue". Subjects remained in their rooms.				

If you click on "Save As..." you will be able to save the log formatted as an HTML document. It can then be loaded into and printed by a variety of utilities and word processors.

You can leave this window visible all the time if you like. It will be automatically updated whenever a new note or modification is made.



## 8. Failure and Recovery

We noted in the introduction that MECMS can survive a variety of failures. In designing this feature we wanted to meet the following goals:

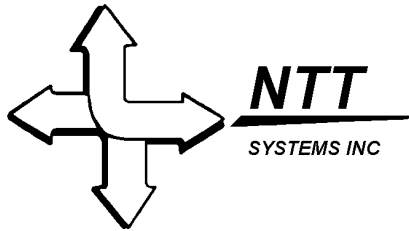
- 1) As a first priority, the system will continue to dispatch work in an uninterrupted manner to all subjects not directly affected by the failure.
- 2) The system should be able to recover from either the failure of the experimenter's console or one or more concurrent workstation failures.
- 3) The recovery mechanism should be easy to use and reliable.

The implementation provides the following features...

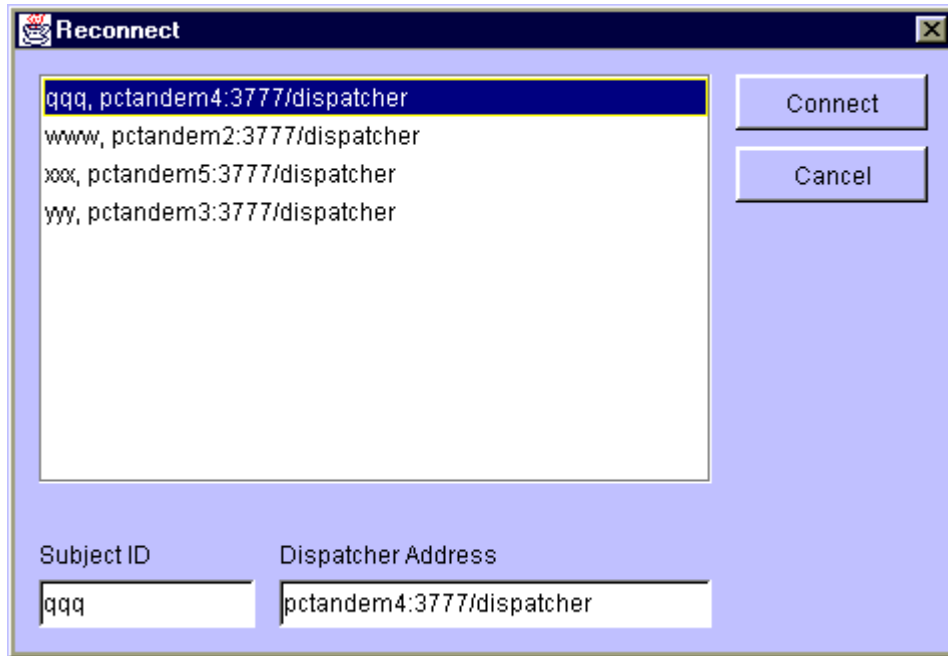
If the console fails, all workstations will continue dispatching tasks to subjects in the normal way. They will not be aware of any change in the environment. If they reach a \*HOLD, they will remain held until the console is restarted.

No workstation failure will affect any other workstation or the operation of the console. An exception to this will arise if a task requires the cooperative actions of the subjects. An example is the TITAN team decision-making task. This task will simply fail to operate if all subjects are not available. The "Suspend", "Kill Task" and "Resume" features are used to get past this condition. The result is missing data points. In MECMS/MpcSusops, the dispatchers manage trial numbers, so trial numbering is properly maintained over failures.

If a workstation fails, it can be rebooted and restarted. All parameters will be restored to the workstation property sheet, and the experimenter simply has to click on the "Restore Workstation" button. Full MECM state information will be passed back to the dispatcher and it will resume in the protocol at the task following the one that was running at the time of failure. Note that MECM can not assume responsibility for information maintained internally or in files by the tasks themselves.



If the console fails, it can be rebooted and restarted. You can initiate recovery by selecting the Dispatcher->Reconnect menu item. The following window will be displayed...



The connection to each dispatcher is reestablished by clicking on the workstation in the top window and clicking on the “Connect” button. This is done for each in turn. The connection process takes place without disturbing dispatching activities on the running workstations. State information is passed back to the console and the workstation displays will appear as though there has been no failure. In particular, all entries will reflect correct task start times and the system will remember which upcoming tasks were manually cancelled by the experimenter.

## 9. Conclusion

MECMS has been used successfully for several experiments at DCIEM. It has proven itself to be reliable and easy to use. It is regularly configured and run by research assistants without management or programmer intervention.

The system can be easily modified to support different task batteries.