

Northwoods Software **SalesWin™ ACT! Add-ons** **Command Tool** **User's Guide**



Command Tool runs on ACT! 2000, 5 or 6 and runs on any version of Windows supported by ACT!.

April 2004
Northwoods Software Corporation
Nashua, NH

6/2003 First Version of Manual

4/2004 Updated to reflect changes up through 2.6

(Check Online Support button in Command Tool for possible updates after 2.6)

Table of Contents

1. Overview	4
2. User Interface to Command Tool	5
Run (Alt R)	5
Settings (ALT S)	5
Cancel (ESC)	6
Add New (ALT A)	6
Remove (ALT E)	6
Rename	6
Online Support (ALT O)	6
Getting Started	6
3. Command File Wizard	7
Step 1 – specify a file name	7
Step 2 – enter a description and type	7
Step 3 – specify the database to use	8
Step 4 – Define Contacts to Use	8
Step 4b – Define a Lookup	9
Step 4c – Define a Query	9
Step 5 – Choose Columns to Output	10
Step 6 – Define Output	11
Step 7 – Wizard Complete	11
4. Command Line Reference	13
General Commands	13
Comment	13
Desc	13
Database Path; username / password	13
ShowProgress TRUE or FALSE	13
Lookup and Query Commands	13
Date Substitutions	14
Interactive Prompts	14
Lookup Field Name,Search String,Lookup Mode Field	15
Query (query)	15
AppLookup	15
Output / Export Commands	15
Output file	16
HTMLOutput file:title	16
Columns Contact Field[,Contact Field...]	16
Headers [TRUE or FALSE]	16
HeadersTitle Header1[,Header2...]	17
Sort field1_dir1,field2_dir2,field3_dir3	17
Append file;text	17
Widths width1,width2,width3	17
Action Commands	18
Open file	18
Runbat file	18

Launch	18
Update Database Commands	18
Set Field1=Value 1;Field 2=Value 2	18
Copy Field 1=Field 2;Field 3=Field 4	19
Swap Field 1=Field 2;Field 3=Field 4	19
AddNote type;record owner;date;text	19
Experimental	19
AppOpen db_path;cly_path;gly_path;username	19
5. Scheduling Command Files with Windows Scheduler	21
6. Understanding ACT! Application Interaction	22

Figures

Figure 1 Command Tool Main Dialog	5
Figure 2 Add New Command File Created?	6

1. Overview

Command Tool provides a way of automating many tasks with ACT!. As the name implies, it is a Tool for building solutions rather than being a solution to a specific problem by itself.

Command Tool has commands that are use both in **Offline** (Batch) mode and **Online** (active ACT! application) mode. The offline mode will open a database using the full path to that database and can operate without ACT! running. The online commands can interact directly with the running ACT! application, using the database that is open and using and setting the current lookup.

Command Tool can

- Perform Lookups and Queries
- Export Contact Data in CSV (text) format
- Create Contact List Reports in HTML
- Automated Edit/Replace (Set, Copy and Swap)
- Current Date relative Saved Query Lookup

In order to perform these actions, Command Tool executes a simple command line syntax. There are commands for opening the database, creating lookups, exporting, reporting, setting contact fields within lookup defined in Command Tool.

The commands live in a text file that is opened, read and executed by Command Tool. Command files that do not explicitly require ACT! to be running can be run in batch mode and can be scheduled to run via the Windows Scheduler.

Here is a simple example of a command file that creates a lookup in ACT! of the contacts that were created today.

```
Database Current();/  
Query "Create Date" = "Today()"  
AppLookup
```

Note that this is basically what a saved query does, but saved queries cannot substitute today's date into the query. There is no way an ACT! Saved query can perform the magic of this simple command file.

2. User Interface to Command Tool

The primary interface to Command Tool is a simple dialog with a list box with some buttons.

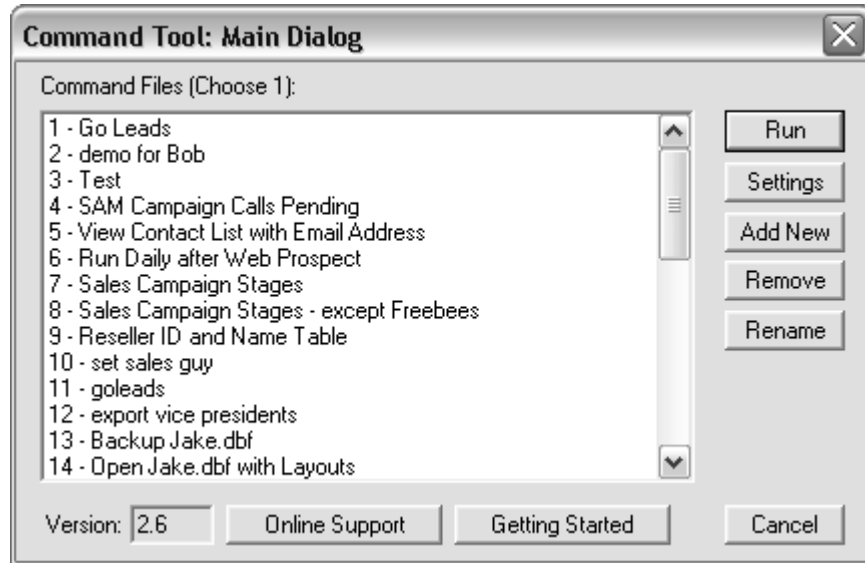


Figure 1 Command Tool Main Dialog

Run (Alt R)

Select a command file from the list and click Run to execute a command file.

You can also run a command file without seeing this dialog by running the program with the id of the Command File you want to run. For example, the command

```
“C:\Program Files\Northwoods Software\CommandTool\CommandTool.exe” 6
```

will run the “6 – Run Daily after Web Prospect” Command File in the screenshot above. Once a command has been run, Command Tool will exit.

Settings (ALT S)

Select a command file from the list and click Settings to see the command file and have a chance to edit it.

Select Edit with Notepad to simply edit the command file in the default editor for .txt files (usually Notepad). Select Command Wizard to be prompted through the editing steps. Note that the Command Wizard only edits what is there, you are not given the same choices as when running the Command Wizard from “Add New”.

Cancel (ESC)

Cancel will exit the tool.

Add New (ALT A)

Add a new command file to the list. Add New will prompt:

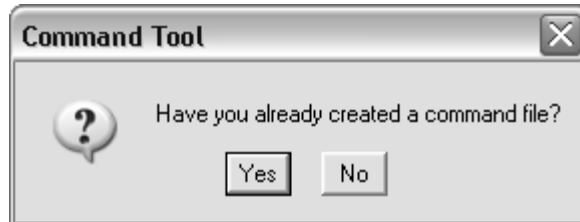


Figure 2 Add New Command File Created?

If you already have the command file ready to use, then click Yes here. Click No to run the Command File Wizard.

Clicking **Yes** here will result in Command Tool prompting for a .TXT file with the File Open Dialog. Once a file is selected, then it will be displayed in the Settings Dialog.

Clicking **No** will start the Command File Wizard.

Remove (ALT E)

Select a command file from the list and click Remove to remove a command file from the list. When a command file is removed, the remaining commands do not get renumbered. This command number will be used the next time you click "Add New".

Rename

Allows renaming of a command.

Online Support (ALT O)

Clicking Online Support will open a web page at Northwoods Software. It will tell you what the current version of Command Tool and other FAQ and usage tips.

Getting Started

Plays a Windows Media video that shows the "Smart Export" video (the old name of Command Tool).

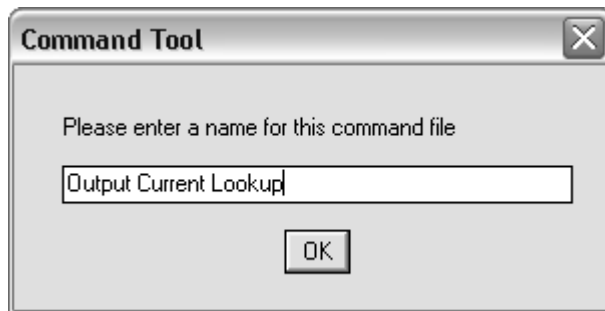
3. Command File Wizard

The Command Wizard can be used to create 3 basic forms of command files.

1. Output (csv or html)
2. Edit / Report (using Set command)
3. Lookup in Application

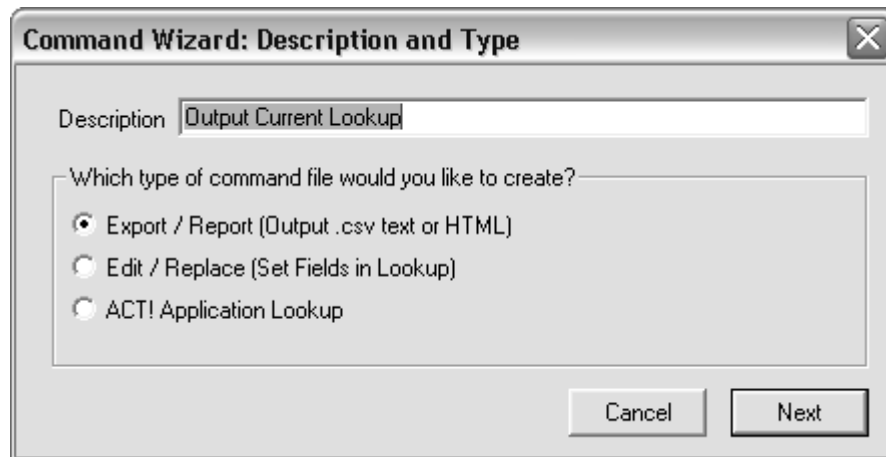
Most command files start with the specification of a Database, followed by a command (Lookup or Query) that specifies which contacts to operate on. Then you make a choice as to whether you want to create a command file of one of the 3 basic types above.

Step 1 – specify a file name



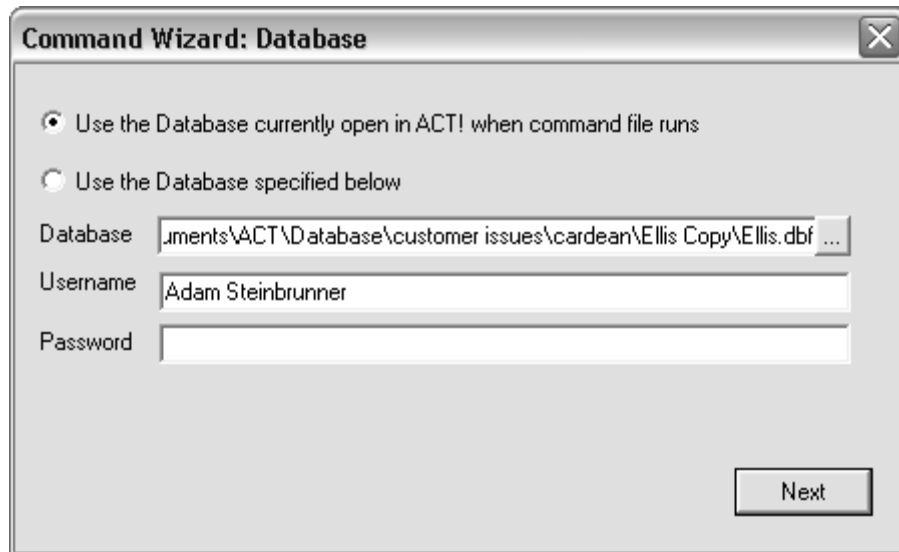
Enter a valid file name for this command file. (no extension needed)

Step 2 – enter a description and type



Choose the type of command file you want to create. Description is saved in the command file, you can type anything you want here.

Step 3 – specify the database to use



This dialog is for is for the Database command line. You can choose to use the database currently open in ACT! when the command file runs, or you can specify a fixed database. Note that for scheduled tasks, you should specify a database. ACT! does not need to be running if you choose “the Database specified below”.

Note in this example, we plan to chose Current Lookup, so it makes sense to choose Current Database.

Make sure ACT! is running and has the proper database open if you chose “currently open” here.

Step 4 – Define Contacts to Use



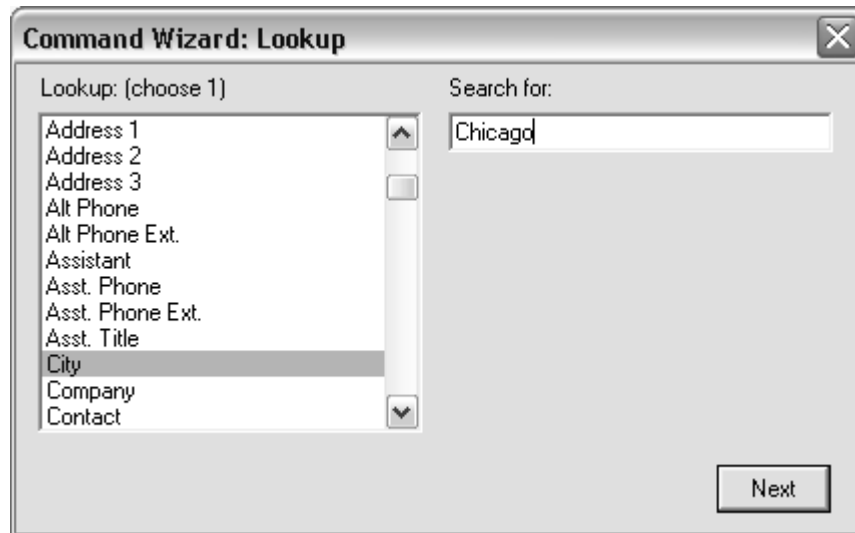
If you select **Current Contact** or **Current Lookup**, you are done on this part.

If you specify **Lookup on Contact Field**, go to Step 4b. If you specify **Advanced Query**, go to Step 4c.

Step 4b – Define a Lookup

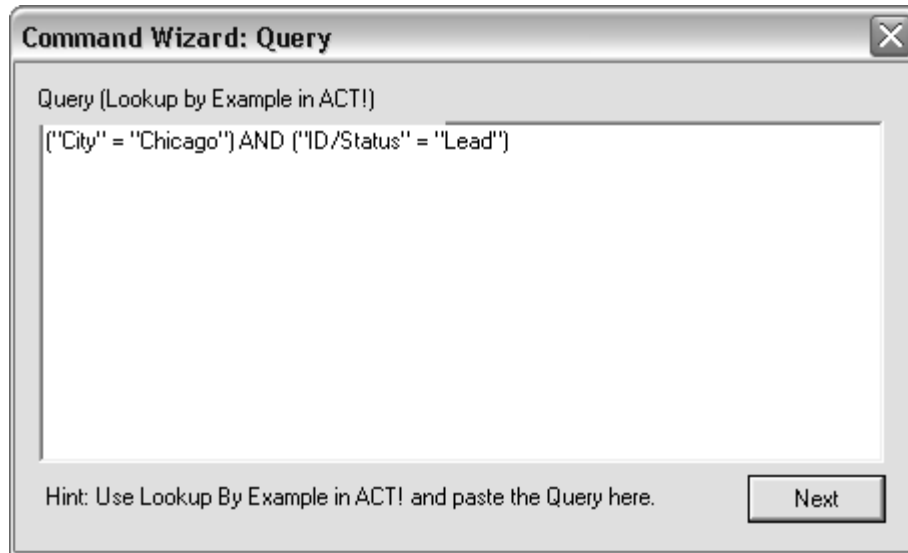
This is a Contact Lookup, just like in ACT!. It defines a Lookup based on a single field starting with a single value.

e.g. City = Chicago

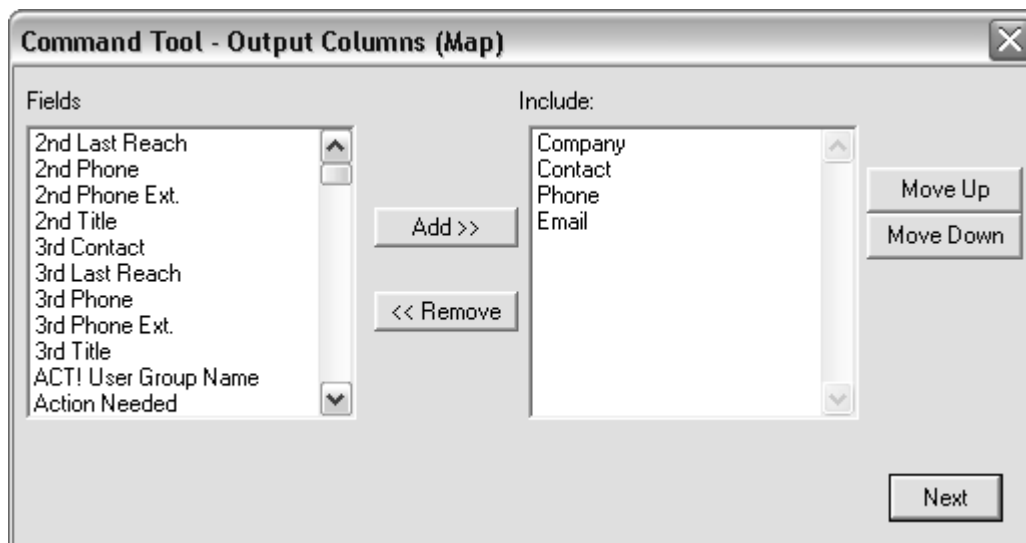


Step 4c – Define a Query

We recommend that you use Lookup By Example in ACT!, and then use **Convert to Advanced Query**. Then use Copy / Paste to copy the query text to this window. Note that ACT! itself does not support Today(), but that you can substitute that for a date when you are defining a Query for Command Tool. See the Lookup and Query Commands section for more detail.



Step 5 – Choose Columns to Output



If you are doing a CSV or HTML output, you will see this dialog:

This dialog defines the Contact Fields that will be output, and the order of the columns.

Note the **Move Up** and **Move Down** buttons loose track of the selection when you use them (something we can't fix because of the development tool we use for this addon).

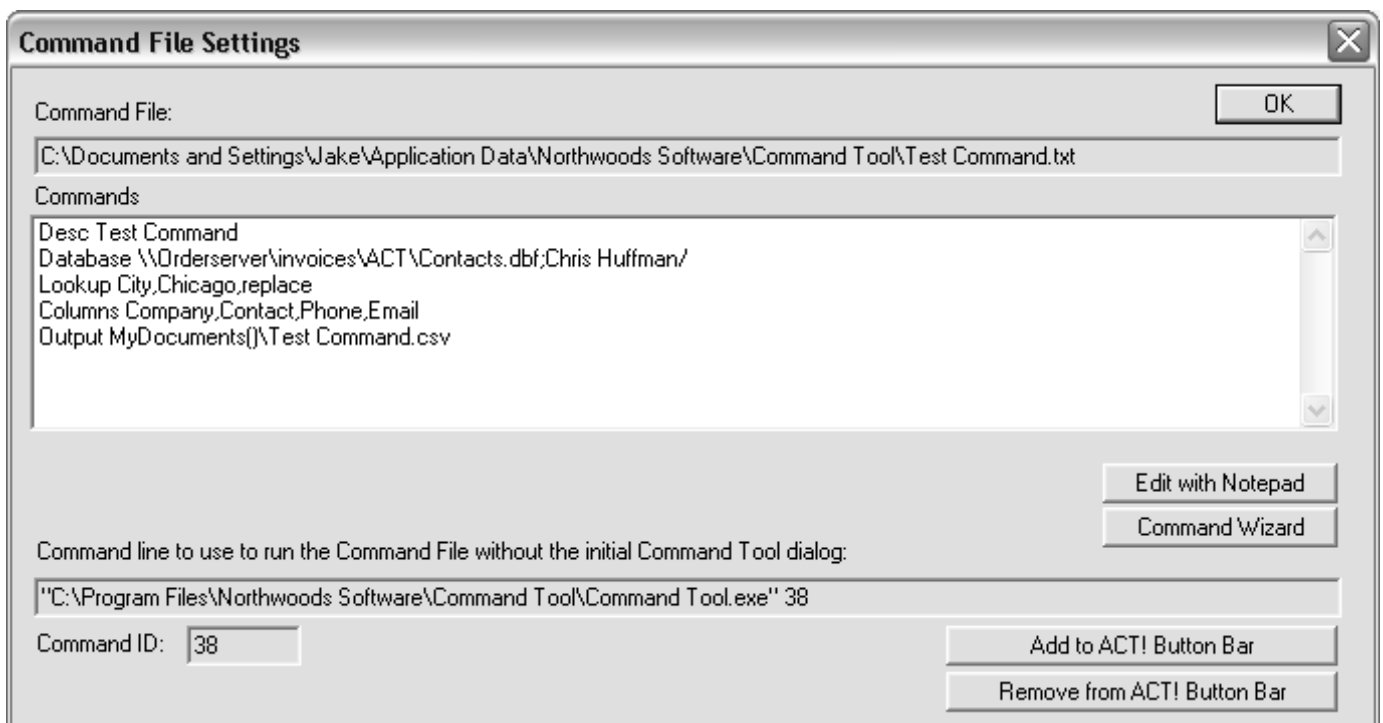
Step 6 – Define Output



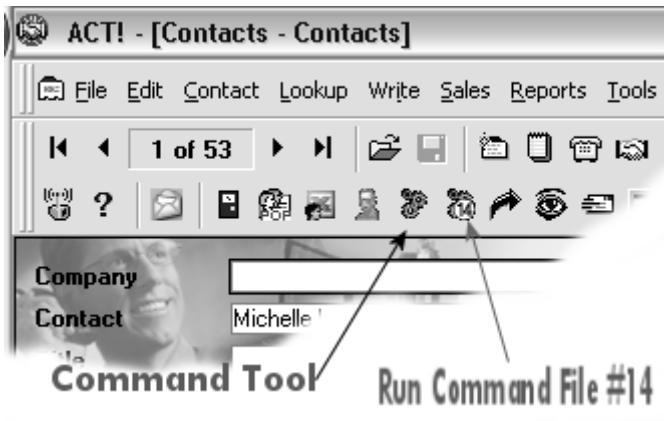
This final dialog chooses CSV (text format) or HTML, as well as output options.

If you check the “Open File” checkbox, the command file will open the file after it is written.

Step 7 – Wizard Complete



You can add command file buttons to the button bar from the Command File Settings Dialog.



4. Command Line Reference

Examples are shown in blue.

General Commands

Comment

Any line starting with an exclamation is a comment.

`! This is a comment.`

Desc

Description. Not currently used, but may be used as optional display text in the Command Tool interface in the future.

`Desc Find all new leads and export them for Bill`

Database Path; username / password

Path: the full path to an ACT! .DBF file

`Database c:\my documents\act\database\contacts.dbf;username/password`

`Database Current();username/password`

Current() will ask ACT! for the currently open database.

ShowProgress TRUE or FALSE

By default, there is no progress window displayed when Command Tool is running. By including a ShowProgress TRUE command, all commands after that will be displayed in the status window as they are executing.

Lookup and Query Commands

The Lookup and Query commands both support special substitutions and actions that occur at runtime. These are:

- Today()
- ThisMonth()
- LastMonth()
- Me()

- Prompt()

Date Substitutions

Command Tool supports Today(), ThisMonth() and LastMonth() as part of the Lookup and Query commands. Today() also supports plus or minus a number of days. These substitutions are made relative to the current date when the command file is run. For the date 6/19/2003 (June 19th, 2003), this table represents what would be substituted....

Today()	6/19/2003
Today()-7	6/12/2003
Today()+2	6/21/2003
LastMonth()	5/*/2003
ThisMonth()	6/*/2003

Interactive Prompts

You can have Command Tool Prompt before performing a Lookup. This allows you to create a single command file that will perform an operation for a lookup you define when you run it.

The syntax of the Prompt() substitution is:

Prompt(Label;Default Value;Description of prompt)

[Lookup=Company,Prompt\(Company;HP;Enter a company name\),replace](#)

Will result in the following prompt

Here, the user can type any company name, and the lookup will be done for Company = “whatever they type”.

Note: The Command Wizard doesn't currently support creating these Prompt fields.

Lookup Field Name,Search String,Lookup Mode Field

Name: The contact field name

Search String: The characters to search for

Lookup Mode: one of Replace,Add or Narrow (these correspond to the ACT! search modes)

You can issue more than one Lookup, just like you can within ACT!, for example:

```
Lookup Title,vice,replace  
Lookup Title,pres,add  
Lookup Title,ceo,add
```

These 3 Lookup lines with Lookup all titles starting with "vice", "pres" and "ceo" in your database.

Note: The Command Wizard doesn't currently support creating more than 1 Lookup line, you will have to add these using a text editor.

Query (query)

Query Command: A standard ACT! Query such as those created by Lookup By Examples within ACT!.

```
Query (Company = "A"*) or (Company = "B"*)
```

This query will lookup all contacts with a Company starting with A or B. Query also supports the Interactive Prompting discussed in Lookup Field Name,Search String,Lookup Mode Field section.

AppLookup

No parameters. Creates a Lookup in ACT! of the contacts defined by the current Lookup or Query command.

```
Database current();/  
Query ("City" = "Chicago") AND ("State" = "Illinois")  
AppLookup
```

Output / Export Commands

Output, **Append** and **HTMLOutput** are commands that write the current lookup to a file.

Columns, **Headers**, **HeadersTitle** are commands that appear before the output commands and modify the output .

Output file

The "file" parameter specifies the name of the file to be written with the output. Output is written in Excel compatible comma separated value (".CSV") format. If the file already exists, it will be overwritten with no warning.

Output `c:\temp\Sample.csv`

If you want a unique output file name each time, then do this:

Output `c:\temp\Sample DateTime().csv`

And the current Year-Month-Day-Hours-minutes-seconds will be added to the file name.

Another substitution supported in **MyDocuments()**.

Output `MyDocuments()\Sample DateTime().csv`

This is useful if you are creating a command file that has to run on multiple systems. Note that both `DateTime()` and `MyDocuments()` are case sensitive currently.

HTMLOutput file;title

The "file" parameter specifies the name of the HTML file to be written with the output. Output is written in HTML format with the columns written in Table format. If the file already exists, it will be overwritten with no warning. `DateTime()` and `MyDocuments()` also work here.

HTMLOutput `c:\temp\Sample.htm;You can put a title here`

HTMLOutput `MyDocuments()\Contact Report DateTime().htm`

Columns Contact Field[,Contact Field...]

Columns `Company,Contact,City,Title,Email,Fax`

The Columns command line set the contact fields that will be exported in the order they columns will be output. Note that these field names are case sensitive. For example, "Company" will work and "company" will not.

Headers [TRUE or FALSE]

Use Headers FALSE to turn off column headers. Can appear in any location before the Output line. (defaults to TRUE)

HeadersTitle Header1[,Header2...]

HeadersTitle allows use of column headers that are different than the actual Field Names.

```
Columns Company,Contact,City,Title,Email,Fax
HeadersTitle Company,Name,Location,Position,E-Mail,Fax #
```

This command has to appear after the Columns command.

Sort field1,dir1,field2,dir2,field3,dir3

Sorts the current lookup or query. Direction can be “up” (ascending) or “down” (descending).

```
Sort Company,up,City,up,Contact,down
```

.This command should appear before any output command

Append file;text

The Append option is supported for use with the Output command. Don't use Append with HTMLOutput.

The Append command allows you to append lines to the output file after it has been written. The "file" parameter should match the one given in the "Output " command line. The number of contacts written to the file can be substituted in the line in the location of %ld.

```
Append c:\temp\Sample.csv;there are %ld contacts in the export
```

Widths width1,width2,width3

The Widths option supported by the Output command.

Warning: this command was added for a specific Western Union import data file. It is very unlikely you will need this command. Fixed field widths like this haven't been seen much since the days of punched cards.

This command can be used to set fixed widths for each column in the export. The number of programs that require fixed width columns is rare, so it is unlikely you will need this command. Note the number of widths in the Widths line must match the number of columns in the Columns line.

```
Columns Company,Contact,City,Title,Email,Phone
Widths 20,30,12,15,40,12
```

Action Commands

Open file

This command line will cause the file to be opened in the Windows default application for that file type. For example, a .txt file would most likely be opened in Notepad, and a .csv file would most likely be opened in Excel.

Open c:\temp\Sample.csv

Open

The Open command without any parameter will open the last file created by Output or HTMLoutput. You need to use this form if you are using DateTime().

Runbat file

This command can be used to run a DOS .bat file after the export. This can be used to do other processing after the export, such as FTPing the file to an Internet location.

Launch

Run an external program with Contact field values as parameters, one run for each contact in the lookup. Uses the Columns Command line to set the launch parameters

```
Database Current();/  
Lookup CurrentLookup()  
Columns Company,Contact,Phone,Email  
Launch C:\program\Launchtest.exe
```

Each parameter is passed “quoted” if necessary, with a space in between each parameter.

Update Database Commands

Set Field1=Value 1;Field 2=Value 2

Allows an Edit/Replace style operation on the Lookup or Query that has been executed. Note: The Set command invalidates the current Lookup. (This is because the Set command may change values in contacts that were used to define the Lookup in the first place.)

```
Database current();/  
Query (“City” = “Chicago”) AND (“State” = “Illinois”)  
Set State=IL
```

Copy Field 1=Field 2;Field 3=Field 4

Allows an Edit/Replace Copy operation on the Lookup or Query that has been executed. Note: The Copy command invalidates the current Lookup. (This is because the Copy command may change values in contacts that were used to define the Lookup in the first place.)

Desc Copy “Sold” to ID/Status History field.

Database current();/

Query (“ID/Status” = “Sold”)
Copy ID/Status History=ID/Status

Swap Field 1=Field 2;Field 3=Field 4

Allows an Edit/Replace Swap operation on the Lookup or Query that has been executed. Note: The Swap command invalidates the current Lookup. (This is because the Swap command may change values in contacts that were used to define the Lookup in the first place.)

AddNote type;record owner;date;text

Adds a Note/History item to the current lookup.

Where type is one of: note, meeting held, meeting not held, to-do done, to-do not done, call left message, call attempted, call complete.

And Record Owner is one of the defined Users in the database.

AddNote Note;Chris Huffman;Today();Followup 1 completed

Experimental

The commands in this section have been added recently, and we have limited experience with them. They aren’t supported by the Command Wizard. You are free to try them and let us know how they work, but don’t be surprised if there are issues...

AppOpen db_path;cly_path;gly_path;username

AppOpen C:\ACT\Database\db.dbf;C:\ACT\Layout\default5.cly;
C:\ACT\Layout\Account52.gly;Chris Huffman

Requires ACT! to be running.

AppOpen will instruct ACT! to close the current database, and open the database specified on the command line with the appropriate Contact and Group Layouts, and finally the appropriate username. If a password is required, ACT! will prompt for it.

Note the command and parameters must be on a single line.

5. Scheduling Command Files with Windows Scheduler

You can run a command file without seeing this dialog by running the program with the id of the Command File you want to run. For example, the command:

```
"C:\Program Files\Northwoods Software\CommandTool\CommandTool.exe" 6
```

will run the command file with id number 6.

The Windows Task Scheduler appears in different places in different versions of Windows. On Windows 98, you will find it in Windows Explorer under "My Computer" with the name "Scheduled Tasks", in the same place you find your "C:" drive. On Windows XP, you will find it in the Start > Settings > Control Panel.

Click **Add Scheduled Task**.

The "Scheduled Task Wizard" will run. Click **Next**.

A dialog with a list of programs will appear. **Pick** Command Tool and click **Next**.

Type a name for the task (something that will help you remember what it does).

Select a **Frequency** (daily, weekly, etc) to run the task and click **Next**.

Select a **Time** (and other time options) to run the task and click **Next**.

If prompted, enter any Windows login info needed and click **Next**

Select **"Open Advanced Properties..."** checkbox and click **Next**.

The Task Properties dialog will open. Go to the end of the "Run" text entry and type a space followed by the number of the command file you want to run.

<Need Command Tool Scheduled Task Screenshot>

Commands not supported in Scheduled Tasks

The following commands require ACT! to be running, and as such probably shouldn't be used in scheduled tasks.

- Database with the Current() parameter
- AppLookup
- Lookup with CurrentLookup() or CurrentContact()

6. Understanding ACT! Application Interaction

Command Tool was originally conceived as an export tool that interacted directly with the ACT! database via the SDK. As such, ACT! did not need to be running to run a Command Tool command file. However, over time, we have decided that some limited interaction with the ACT! Application can be useful.

However, most of Command Tool is still faithful to this original notion of use, and the commands run within Command Tool with little need for ACT! itself.

The following commands DO require ACT! to be running.

Database Current()

Lookup CurrentLookup()

Lookup CurrentContact()

AppLookup

7. Samples

Below are some ideas about how Command Tool can be useful in your office.

Assign Owner (sales person) based on Region

The following assigns leads that were entered today to 1 of 2 sales people. Note there Command Tool can execute more than one Query in single command file. (The wizard can't create these, but you can with Notepad.)

Note that long queries must really be a single line of text... wrapping is not supported by the Command Tool. We recommend using the Query Tool (Lookup by Example) in ACT! and then use Copy/Paste to the command file.

Command file: Assign Sales Person for new Leads.txt

```
Database Current();/
!
! Assign the New England states to Jeff Nelson
!
Query ("Create Date" = "Today()") AND ("State" =
"ME"OR "State" = "NH" OR "State" = "VT" OR "State" =
"MA"OR "State" = "CT" OR "State" = "RI")
Set Record Manager=Jeff Nelson
!
! Assign PA and NY to Chris Huffman
!
Query ("Create Date" = "Today()") AND ("State" = "NY"
OR "State" = "PA")
Set Record Manager=Chris Huffman
```

Output Current Contact to CSV file with unique filename

Command file: Output Current Contact to CSV file with unique filename.txt

```
! Open the database open in ACT! app
Database Current();/

! Lookup the current contact in ACT! app
Lookup CurrentContact()

Columns Company,Contact,Phone,Email,Web Site
Output c:\temp\contact DateTime().csv
```

Lookup Today's Leads

Command file: Todays Leads.txt

```
Desc Lookup Created Today
Database Current();/
Query "Create Date" = "Today()"
AppLookup
```

Create Contact List

Command file:Contact List.txt

```
Desc View Contact List with Email Address
ShowProgress TRUE
Database Current();/
Lookup CurrentLookup()
Columns Company,Contact,Phone,Email
HTMLoutput MyDocuments()\ContactList DateTime().htm;
Open
```

This command file creates a “Contact List” in a web browser. If you try to create a contact list with the Email field inside ACT!, the performance is awful. This command also “saves” the current lookup so you can refer to it latter. (Although you can’t restore it in ACT!, you can click on the contact names and “go to” the contact you want to see.)

This command file opens the “Current” database open in ACT!, then reads the Current Lookup in ACT!, then creates an HTML file. It writes the file to your “My Documents” folder and includes a Date/Time stamp in the file so it won’t overwrite an earlier contact list created with this command file. The final operation is to open the file in your default browser.

<End>