

## **InContact! Utility Extensions**

Developers can create their own utilities that will work with InContact. Any extensions you add can be easily included in the menu structure of InContact allowing seamless integration. You can write the extension in most any language that produces an EXE, COM or BAT file to execute. You can include copyright and shareware information which will be presented to the user each time they run the application. Two example extensions are included with this version of InContact: one rebuilds and repairs files and the other simply calls up a notepad file to allow the user to keep global notes for each contact file. The latter required no programming because InContact passes the filename to the specified program in the command line buffer. In addition, InContact can optionally pass the following information to your program: current record number, current file and user-selected filename. Add to this full DDE support and the normal power of Windows and you can create most anything.

Here are some possible ideas:

- Links to the Windows accessories (calendar, write, etc.)
- File imports from other programs
- Additional database applications that can key off a contact (more specialized information)
- Enhanced functionality to augment existing InContact operations
- A Client billing application

To help you get going, we have included a sample utility extension "Personal.EXE" as well as the complete specs necessary to make integrated utility extensions for InContact.

## **FILE FORMATS:**

### **"filename.CNT" Contact file (Random Access)**

This file contains the actual data for each of the contacts. By taking the record number passed to your extension by InContact, you can read current contact information. We strongly recommend that you do not change any of this information and that you only use it for a reference.

```
'-
' Record Structure for Contact File (*.CNT)
'-
Type ContactRecord
  ContactName      As String * 40 ' Contact Name (KEY)
  Category         As String * 15 ' Category
  Status          As String * 15 ' Status
  BusPhone        As String * 25 ' Phone Numbers ...
  FaxPhone        As String * 25
  MobilPhone      As String * 25
  ModemPhone      As String * 25
  HomePhone       As String * 25
  OtherPhone      As String * 25
  Firm            As String * 40 ' Work Address Info ...
  FirmAddress     As String * 40
  FirmCity        As String * 25
  FirmState       As String * 25
  FirmZip         As String * 15
  Home            As String * 40 ' Home Address info ...
  HomeAddress     As String * 40
  HomeCity        As String * 25
  HomeState       As String * 25
  HomeZip         As String * 15
  LastCall        As Double                ' VB DateValue
  CallBack        As Double                ' Ditto
  Extra1          As Long                  ' RESERVED
  NoteLink        As Long                  ' Notes Link
  LogLink         As Long                  ' Log Rec Link
  OptLink         As Long                  ' Option Link (attached files)
  BirthDate       As Double                ' Age is auto-calculated
  Flags           As String * 1            ' Compressed into a single byte
  Extra2          As String * 20 ' RESERVED
End Type
```

### **"filename.NDX" Index file (ASCII)**

This ASCII file is a formatted list of contacts for a given file. Each line consists of a contact name followed by two tabs (ASCII value of 9) and the record number. The list does not need to be kept sorted (that is done at load time) and can be loaded directly into a listbox. You must be VERY CAREFUL if you modify this file in any way. In most cases, you will probably only WRITE to the selection list (filename.SEL).

### **"filename.SEL" Current selected contacts (ASCII)**

This file is arranged in the same way as the NDX file but contains only contacts selected by the user. If you wish to perform special selects yourself, you can write this file so that InContact can allow the user to use your selection list (this list is auto-reloaded if changed since the last time the Locate dialog was called).

### **"filename.LOG" Log entries (Random Access Linked List)**

This random access file handles several entries per contact. To do so, each entry tracks the record number of the contact they relate to. In addition, they are linked together via the NextRec and PrevRec numbers that allow you to walk from the first item for a contact through the last. This is accomplished by using the following logic:

1. Get a contact record
2. Get the record number in the LogLink element
3. If the LogLink record number is not zero,
4. Locate that record number in the log file
5. Read the data for that record
6. Read the NextRec element
7. If the NextRec record number is not zero, loop back to 4.

```
'_
' Record structure for log file (*.LOG)
'_
Type LogRecord
  Description      As String * 40 ' Description
  Type             As Integer           ' 0=Phone, 1=letter, 3-5=RESERVED
  Start           As Double             ' Start date/time stamp
  Stop            As Double             ' End date/time stamp
  Results         As String * 8        ' Encoded results
  LinkRec         As Long               ' Contact File Record Number
  PrevRec        As Long               ' Previous log item for this contact
  NextRec        As Long               ' Next log item for this contact
End Type
```

### **"filename.NOT" Notes (Random Access Linked List)**

This file contains all the user notes. Each record has 100 bytes of data that must be put together by stepping through the linked-list (the same process for the LOG files). This allows all the notes for all the contacts to be placed in a single file and still allow up to 64k.

```
'_
' Record structure for Notes (*.NOT)
'_
Type NoteRecord
  InfoBlock       As String * 100      ' Block of note text
  LinkRec         As Long              ' Contact File Record Number
  PrevRec        As Long              ' Previous note block
  NextRec        As Long              ' Next note block
End Type
```

### **"filename.LNK" Attached file list (Random Access Linked List)**

This is a list of files attached to different contacts. And it is also set up the same way as the LOG and Note files.

```
'_
' Record structure for file attach (*.LNK)
'_
Type FileLinkRecord
  Description      As String * 30 ' Description
  FileName         As String * 128      ' Filename + path
  LinkRec          As Long              ' Contact File Record Number
  PrevRec          As Long              ' Previous link for this contact
  NextRec          As Long              ' Next link for this contact
End Type
```

### **"filename.CAT" Custom categories (ASCII)**

This file is simply an ASCII list of user defined categories. Each item is on its own line.

### **"filename.STS" Custom status keywords (ASCII)**

This is also an ASCII list with each item listed on its own line.

### **Reading Parameters from InContact:**

When your program is called, InContact will supply several parameters in the command line buffer. These include current file, record number and any other information you request in your extension definition flags (see the text on CONTACT.EXT file format above). You can read these parameters in a variety of ways (look up something like "Command Line Arguments" in your language manual):

In C, use the ARGV and \*\*ARGV as parameters in your main()  
In BASIC you can usually read the COMMAND\$ and parse out the values yourself  
From a batch file, the %1 %2 %3 etc. parameters allow you to use them as variables.

InContact will send you the following information in the following order:  
drive:\path\filename /R:nnn /S:n /I:"nnn"

Where the filename will contain a complete path and a file with no extension (UNLESS you force one with the /E definition flag explained in the CONTACT.EXT text above -OR- if you have the user specify a filename with the /F definition flag).

The other parameters specify the following:  
/R:nnn Record number (nnn)

The proper use of these command line parameters depend on InContact having a clear idea of what you want. You need to carefully read the text on CONTACT.EXT above to have a better understanding of what InContact expects.

### **DDE SPECS:**

InContact also acts as a DDE Server. If you are familiar with DDE, you should easily be able to link to the data fields in the contact window with some word processors, EXCEL or your own Windows application (Visual Basic supports DDE). The Application ID is CONTACT and the Link Topic is Contact (redundancy is great). The following is a list of all the available Link Items in InContact:

Contact,	WorkPhone,	FaxPhone,	HomePhone,	MobilPhone,
ModemPhone,	OtherPhone,	Firm,	FirmAddress,	FirmCity,
FirmState,	FirmZip,	Home,	HomeAddress,	HomeCity,
HomeState,	HomeZip,	LastCall,	CallBack,	Birthday,
Age,				

Special DDE Link Items:

DDE_RecordNumber	Allows your application to be aware of the current contact record number.
DDE_CurrentFileName	Contains the current file in use (without file extension)
DDE_Control	Sends messages to your application:-1=Add New, -2=Cancel, -3=Store, -4=Close.

If you create any extensions, by all means distribute them and send a copy our way. We keep them on file so that we can recommend them to users. If you like, you can let us distribute them for you; we are very easy to work with.