Telix DDE Server

Telix for Windows provides a DDE server that may be used by any application that supports DDE conversations, including SALT scripts themselves. All examples given in this document will be fully functional SALT scripts.

This document assumes you understand the concepts of Dynamic Data Exchange. It is beyond the scope of this document to explain the workings of DDE as implemented in Windows. You can find all of the information presented here in the Telix help file by searching for the keyword phrase "Dynamic Data Exchage", but is provided here for users who wish to print it.

The service name for Telix for Windows is **Telix** and it supports three topics: **System**, **Telix**, and **DialingDirectory**.

The following is an outline of the server and the topics and items it provides:

⇒ **Telix**

- **System**
    - ◆ ***Request***
        1. SysItems
        2. Topics
        3. Formats
        4. Status
        5. Help

- **Telix**
    - ◆ ***Execute***
        1. Call <phonebook entry>
        2. ExecScript <filename>
        3. ScriptEditor
        4. Compile <filename>
        5. Upload <protocol,files>
        6. Download <protocol,files>
        7. OpenCFG <filename>
        8. SaveCFG
        9. Activate
        10. Exit
    - ◆ ***Request***
        1. TopicItemList
        2. Connection
        3. WindowState
        4. ConnectDevice
        5. TerminalDevice
        6. ExpenseDevice
        7. TranslateTable
        8. KeyboardMacro
    - ◆

Poke

1. Connection <data string>
2. WindowState <state>
3. ConnectDevice <connect device name>
4. TerminalDevice <terminal device name>
5. ExpenseDevice <expense device name>
6. TranslateTable <translate table name>
7. KeyboardMacro <keyboard name>

- **DialingDirectory**
    - ***Execute***
        1. Open [<filename>]
        2. Save [<filename>]
        3. Append [<filename>]
        4. AddEntry <new entry data>
        5. DeleteEntry <phonebook entry>
        6. Exit
    - ***Request***
        1. TopicItemList
        2. Entries
        3. Filename
    - ***Poke***
        1. Filename <filename>

The System Topic:

The System topic is provided to request data about the **Telix** DDE service. All items in this topic are DDE request items.

Command: SysItems
Description: This item returns the items available in the **System** topic. The items are returned as a tab-delimited string.
Example:

```
main()
{
 int DDEConv;
 str Buff[255];

 DDEConv = DDEInitiate("TELIX", "SYSTEM");
 if (DDEConv > 0) {  // DDE conversation established.
   DDERequest(DDEConv, "SysItems", Buff);
   prints(Buff);

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: Topics
Description: This item returns the topics that are available in the **Telix** DDE service. The topics are returned as a tab-delimited string.
Example:

```
main()
{
 int DDEConv;
 str Buff[255];

 DDEConv = DDEInitiate("TELIX", "SYSTEM");
 if (DDEConv > 0) {  // DDE conversation established.
   DDERequest(DDEConv, "Topics", Buff);
   prints(Buff);

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: Formats

Description: This item returns the clipboard formats that are supported by Telix for Windows. Thie items are returned as a tab-delimited string.

Example:

```
main()
{
 int DDEConv;
 str Buff[255];

 DDEConv = DDEInitiate("TELIX ", "SYSTEM");
 if (DDEConv > 0) {  // DDE conversation established.
   DDERequest(DDEConv, "Formats", Buff);
   prints(Buff);

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: Status

Description: This item returns the current status of Telix. The status will be Ready, Busy, Loading or Exiting. If any value other than Ready is returned, no DDE messages should be sent to Telix except **Status** requests and conversation terminations. When **Status** returns Ready, it is safe to resume DDE transactions with Telix.

Example:

```
main()
{
 int DDEConv;
 str Buff[255];

 DDEConv = DDEInitiate("TELIX", "SYSTEM");
 if (DDEConv > 0) {  // DDE conversation established.
   DDERequest(DDEConv, "Status", Buff);
   printsc("Telix's status is ");
   prints(Buff);

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: Help
Description: This item will provide a brief help message about using the Telix DDE services.
Example:

```
main()
{
 int DDEConv;
 str Buff[255];

 DDEConv = DDEInitiate("TELIX", "SYSTEM");
 if (DDEConv > 0) {  // DDE conversation established.
   DDERequest(DDEConv, "Help", Buff);
   prints(Buff);

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

The Telix Topic:

The Telix topic is used to access items that are either global to Telix or act upon the current connection.

Command: Call
Type: Execute
Description: Use this command to call entries in the PhoneBook and, alternatively, phone numbers passed with this command.
Params: A string containing the entries to be dialed. Entries can be specified as a PhoneBook entry number, a PhoneBook entry name, or partial name, enclosed in pipe (|) symbols, or a manually entered phone number prefaced by an 'm'. If an actual pipe symbol is required, use a double pipe (||).
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   // Call deltaComm Online
   DDEExecute(DDEConv, "Call |deltaComm Online|");
   // Call deltaComm Online, entry #3 and a manual number.
   DDEExecute(DDEConv, "Call |deltaComm|3|m919-481-9399|");

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: ExecScript
Type: Execute
Description: Use this command to execute SALT and SIMPLE scripts.
Params: The filename of the script to be executed.
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   // This script file must be in the scripts directory.
   DDEExecute(DDEConv, "ExecScript EXAMPLE.SLC");
   DDEExecute(DDEConv, "ExecScript C:\MYSCRIPTS\EXAMPLE.SLC");
   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: ScriptEditor
Type: Execute
Params: None.
Description: This item will start the Telix script editor.
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   DDEExecute(DDEConv, "ScriptEditor");
   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: Compile
Type: Execute
Description: This item will instruct Telix to compile a SALT or SIMPLE script.
Params: The filename of the script to be compiled.
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   DDEExecute(DDEConv, "Compile EXAMPLE.SLT");
   DDEExecute(DDEConv, "Compile C:\MYSCRIPTS\EXAMPLE2.SLT");

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: Upload
Type: Execute
Description: This item will cause Telix to begin a file upload.
Params: The protocol to be used (See SALT send() function) followed by the filename(s) to be uploaded.
Example:

```
main()
{
 int DDEConv;
 str Buff[255];

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   // Make sure we are connected.
   DDERequst(DDEConv, "Connection", Buff);
   if (Buff != "") {
     // Upload 2 files using ZModem.
     DDEExecute(DDEConv, "Upload Z,FILE1.ZIP FILE2.ZIP");
   } else {
     prints("Not connected!");

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: Download
Type: Execute
Description: This item will cause Telix to begin a file download.
Params: The protocol to be used (See SALT receive() function) followed by the filename(s) to be downloaded. If you will be using a protocol that sends filenames, you may omit the filename(s).
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   // Receive file using XModem
   DDEExecute(DDEConv, "Download X,MAILPKT.ZIP");
   // Receive file using ZModem and the default filename.
   DDEExecute(DDEConv, "Download Z");

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: OpenCFG
Type: Execute
Description: This item will make a given Telix configuration file active.
Params: The configuration filename to open.
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   DDEExecute(DDEConv, "OpenCFG C:\TFW\OTHER.CFG");

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: SaveCFG
Type: Execute
Description: This item will save the current Telix configuration.
Params: If no parameter is given, the configuration is saved to the current file. Optionally, you may specify a new filename to be used for storing the configuration.
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   // Save configuration to default filename.
   DDEExecute(DDEConv, "SaveCFG");
   // Save configuration to specified filename.
   DDEExecute(DDEConv, "SaveCFG C:\TFW\OTHER.CFG");

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: Activate
Type: Execute
Params: None.
Description: This item will cause Telix to become the active application.
Example:

```
main()
{
 int DDEConv;
 str Buff[255];

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   DDEExecute(DDEConv, "Activate");

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: Exit
Type: Execute
Description: This item causes Telix to close.
Params: None.
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   DDEExecute(DDEConv, "Exit");

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: TopicItemList
Type: Request
Description: This item returns a list of all topic items in the **Telix** topic. The items are returned as a tab-delimited string.
Example:

```
main()
{
 int DDEConv;
 str Buff[255];

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   DDERequest(DDEConv, "TopicItemList", Buff);
   prints("Available Items for the Telix topic are:");
   prints(Buff);

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: Connection
Type: Request
Description: This item returns then name of the system that Telix is currently connected to. If there is no connection, a blank string is returned.
Example:

```
main()
{
 int DDEConv;
 str Buff[255];

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   DDERequest(DDEConv, "Connection", Buff);
   if (Buff != "")
     prints("Connected to " + Buff);
   else
     prints("Not connected.");

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: WindowState
Type: Request
Description: This item returns the state of the Telix window. The possible values are Maximized, Minimized and Normal.
Example:

```
main()
{
 int DDEConv;
 str Buff[255];

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   DDERequest(DDEConv, "WindowState", Buff);
   if (Buff != "Minimized")
     DDEPoke(DDEConv, "WindowState", "Minimized");

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: ConnectDevice
Type: Request
Description: This item returns the name of the connect device that Telix is using.
Example:

```
main()
{
 int DDEConv;
 str Buff[255];

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   DDERequest(DDEConv, "ConnectDevice", Buff);
   prints("Current Connect Device is " + Buff);

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: TerminalDevice
Type: Request
Description: This item returns the name of the terminal device that Telix is using.
Example:

```
main()
{
 int DDEConv;
 str Buff[255];

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   DDERequest(DDEConv, "TerminalDevice", Buff);
   prints("Current Terminal Device is " + Buff);

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: ExpenseDevice
Type: Request
Description: This item returns the name of the expense device that Telix is using.
Example:

```
main()
{
 int DDEConv;
 str Buff[255];

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   DDERequest(DDEConv, "ExpenseDevice", Buff);
   prints("Current Expense Device is " + Buff);

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: TranslateTable
Type: Request
Description: This item returns the name of the translation table that Telix is using.
Example:

```
main()
{
 int DDEConv;
 str Buff[255];

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   DDERequest(DDEConv, "TranslateTable", Buff);
   prints("Current Translation Table is " + Buff);

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: KeyboardMacro
Type: Request
Params: None.
Description: This item returns the name of the keyboard macro table that Telix is using.
Example:

```
main()
{
 int DDEConv;
 str Buff[255];

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   DDERequest(DDEConv, "KeyboardMacro", Buff);
   prints("Current Keyboard Macro Table is " + Buff);

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: Connection

Type: Poke

Description: This item sends data to the terminal. All data sent is translated before passed to the terminal so that special strings such as '^M' can be translated into carriage returns.

Params: The data to be sent to the terminal.

Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   // Reset the modem settings to factory defaults
   DDEPoke(DDEConv, "Connection", "AT&F^M");

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: WindowState

Type: Poke

Description: This item sets the window state of Telix.

Params: The new window state: Maximized, Minimized and Normal.

Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   // Maximize Telix
   DDEPoke(DDEConv, "WindowState", "Maximized");

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: ConnectDevice
Type: Poke
Description: This item sets the connect device that Telix will use.
Params: The name of the connect device.
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   DDEPoke(DDEConv, "ConnectDevice", "ELSA MicroLink");

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: TerminalDevice
Type: Poke
Description: This item sets the terminal device that Telix will use.
Params: The name of the terminal device.
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   DDEPoke(DDEConv, "TerminalDevice", "VT100");

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: ExpenseDevice
Type: Poke
Description: This item sets the expense device that Telix will use.
Params: The name of the expense device.
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   DDEPoke(DDEConv, "ExpenseDevice", "919 Area Code");

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: TranslateTable
Type: Poke
Description: This item sets the translation table that Telix will use.
Params: The name of the translation table.
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   DDEPoke(DDEConv, "TranslateTable", "SPECIAL");

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: KeyboardMacro
Type: Poke
Description: This item sets the keyboard macro table that Telix will use.
Params: The name of the keyboard macro table.
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("TELIX", "TELIX");
 if (DDEConv > 0) {  // DDE conversation established.
   DDEPoke(DDEConv, "KeyboardMacro", "TradeWar");

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

The DialingDirectory Topic:

The DialingDirectory topic is used to access the data stored in a Telix PhoneBook, and to show or hide the Dialing Directory (the window Telix uses to display PhoneBook data).

Command: Open
Type: Execute
Description: This item opens a PhoneBook in the Dialing Directory. The Dialing Directory will be opened if it is not currently displayed.
Params: The filename of the PhoneBook to open. If no filename is given, the Dialing Directory will be displayed with the current PhoneBook.
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("TELIX", "DIALINGDIRECTORY");
 if (DDEConv > 0) {  // DDE conversation established.
   // Open LONGDIST.FBK and display the Dialing Directory
   DDEExecute(DDEConv, "Open C:\TFW\LONGDIST.FBK");

   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: Save
Type: Execute
Description: This item saves the current PhoneBook to disk.
Params: The filename to save the PhoneBook to. If no filename is given, the current filename is used.
Example:

```
main()
{
 int DDEConv;
 str Buff[255];

 DDEConv = DDEInitiate("TELIX", "DIALINGDIRECTORY");
 if (DDEConv > 0) {  // DDE conversation established.
   // Save PhoneBook, make a backup and switch back to original.
   DDERequest(DDEConv, "Filename", Buff);
   DDEExecute(DDEConv, "Save");
   DDEExecute(DDEConv, "Save C:\TFW\BACKUP.FBK");
   DDEExecute(DDEConv, "Open " + Buff);

   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: Append
Type: Execute
Description: This item opens a PhoneBook and adds its contents to the current PhoneBook.
Params: The name of the PhoneBook to be appended.
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("TELIX", "DIALINGDIRECTORY");
 if (DDEConv > 0) {  // DDE conversation established.
   // Add entries in NEWSTUFF.FBK to TELIX.FBK
   DDEExecute(DDEConv, "Open C:\TFW\TELIX.FBK");
   DDEExecute(DDEConv, "Append C:\TFW\NEWSTUFF.FBK");

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: AddEntry
Type: Execute
Description: This item adds a new entry to the PhoneBook.
Params: The data for the new entry in the following format: name,[number,[terminal, [connectdevice,[script,[username,[password,
[comment]]]]]]].
Items enclosed in brackets are optional. Also, items may be left blank by simple not including a value.
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("TELIX", "DIALINGDIRECTORY");
 if (DDEConv > 0) {  // DDE conversation established.
   // Add a minimal entry with just name and number.
   DDEExecute(DDEConv, "AddEntry deltaComm,1-919-481-9399");
   // Add a detailed entry, without a linked script
   // Note the "^".  Used for string continuation.
   DDEExecute(DDEConv, "AddEntry deltaComm Online,1-919-481-9399^
     ,ANSI-BBS,ELSA MicroLink,,Jeff Woods,NotHere,deltaComm ^
     Development Support BBS");

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: DeleteEntry
Type: Execute
Description: This item removes an entry from the PhoneBook.
Params: The name of the entry to be deleted enclosed in quotes (**"**), or the number of the entry without quotes.
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("TELIX", "DIALINGDIRECTORY");
 if (DDEConv > 0) {  // DDE conversation established.
   // Delete the entry named "Defunct BBS"
   DDEExecute(DDEConv, "DeleteEntry ^"Defunct BBS^"");
   // Delete entry number 3
   DDEExecute(DDEConv, "DeleteEntry 3);

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: Exit
Type: Execute
Description: This item closes the Dialing Directory window.
Params: None.
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("TELIX", "DIALINGDIRECTORY");
 if (DDEConv > 0) {  // DDE conversation established.
   DDEExecute(DDEConv, "Exit");

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: TopicItemList
Type: Request
Description: This item returns a list of all topic items in the **DialingDirectory** topic. The items are returned as a tab-delimited string.
Example:

```
main()
{
 int DDEConv;
 str Buff[255];

 DDEConv = DDEInitiate("TELIX", "DIALINGDIRECTORY");
 if (DDEConv > 0) {  // DDE conversation established.
   DDERequest(DDEConv, "TopicItemList", Buff);
   prints("Available Items in the DialingDirectory Topic are:");
   prints(Buff);

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: Entries
Type: Request
Description: This item returns the number of entries in the current PhoneBook.
Example:

```
main()
{
 int DDEConv;
 str Buff[255];

 DDEConv = DDEInitiate("TELIX", "DIALINGDIRECTORY");
 if (DDEConv > 0) {  // DDE conversation established.
   DDERequest(DDEConv, "Filename", Buff);
   printsc("The " + Buff + " PhoneBook contains ");
   DDERequest(DDEConv, "Entries", Buff);
   prints(Buff + " entries.");

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: Filename
Type: Request
Description: This item returns the filename of the current PhoneBook.
Example:

```
main()
{
 int DDEConv;
 str Buff[255];

 DDEConv = DDEInitiate("TELIX", "DIALINGDIRECTORY");
 if (DDEConv > 0) {  // DDE conversation established.
   DDERequest(DDEConv, "Filename", Buff);
   printsc("The " + Buff + " PhoneBook contains ");
   DDERequest(DDEConv, "Entries", Buff);
   prints(Buff + " entries.");

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```

Command: Filename
Type: Poke
Description: This item changes the current PhoneBook. It differs from the **Open** command in that it does not display the DialingDirectory window.
Params: The filename of the PhoneBook to be opened.
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("TELIX", "DIALINGDIRECTORY");
 if (DDEConv > 0) {  // DDE conversation established.
   // Call the deltaComm Online BBS, which is in the LONGDIST.FBK
   // PhoneBook.  Note that this does not display the Dialing
   // Directory when changing PhoneBooks.
   DDEPoke(DDEConv, "Filename", "C:\TFW\LONGDIST.FBK");
   DDEExecute(DDEConv, "Call |deltaComm|");

   // close the DDE conversation.
   DDETerminate(DDEConv);
 } else {  // DDE conversation could not be established.
   prints("Could not connect.");
 }
}
```