Telix Script Editor DDE Server

The Telix for Windows Script Editor 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 the Script Editor is **SEditor** and it supports three topics: **System**, **SEditor**, and **<filename>**. The **<filename>** topic is the name of any open file in the script editor. You can retrieve a list of open files by using the **System** topic and requesting the **Topics** item (this is illustrated in the **System | Topics** example below).

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

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

- **SEditor**
    - ◆ *Execute*
        1. Open <filename>
        2. Save <filename>
        3. Close <filename>
        4. Compile <filename>
        5. Exit
        6. Activate
- **<filename>**
    - ◆ *Execute*
        1. Save
        2. Close
        3. Compile
    - ◆ *Poke*
        1. Code <string>
        2. Cursor <line,column>
    - ◆ *Request*
        1. Code
        2. Cursor

The System Topic:

The System topic is provided to request data about the **SEditor** 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("SEDITOR", "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 **SEditor** DDE service. The topics are returned as a tab-delimited string.
Example:
```
main()
{
 int DDEConv;
 str Buff[255];

 DDEConv = DDEInitiate("SEDITOR", "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 the Script Editor.
              Thie items are returned as a tab-delimited string.
Example:

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

 DDEConv = DDEInitiate("SEDITOR", "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 the Script Editor. The status will be
              Ready, Busy, or Unknown. If Busy, no DDE messages should be sent to the
              Script Editor except **Status**. When **Status** returns Ready, it is safe to resume
              DDE transactions with the Script Editor. If Unknown is returned, a situation
              has been encountered by the Script Editor that it was unprepared for. All DDE
              conversations should be terminated and the Script Editor should be closed and
              restarted.
Example:

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

 DDEConv = DDEInitiate("SEDITOR", "SYSTEM");
 if (DDEConv > 0) {  // DDE conversation established.
   DDERequest(DDEConv, "Status", Buff);
   printsc("The Script Editor is ");
   prints(Buff);
   if (Buff == "Unknown")
     prints("It should be closed now to avoid data loss!");

   // 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 for using the Script Editor DDE services.
Example:

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

 DDEConv = DDEInitiate("SEDITOR", "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 SEditor Topic:

The SEditor topic is used to access items that are either global to the Script Editor or act upon the active file in the Script Editor. All items in this topic are DDE Execute items.

Command: Open
Description: This item opens a file in the editor, or if the file is already loaded, makes it the active file.
Params: The file to be opened or made active. If no file is given, a new untitled file will be created. If a filename is given, but does not exist, the file will be created.
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("SEDITOR", "SEDITOR");
 if (DDEConv > 0) {  // DDE conversation established.
   // Open an existing file.
   DDEExecute(DDEConv, "Open c:\tfw\scripts\example.slt");
   // Create a new, untitled file.
   DDEExecute(DDEConv, "Open");

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

Command: Save
Description: This item saves a file to disk.
Params: The file to be saved. If no file is given, the active file is saved.
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("SEDITOR", "SEDITOR");
 if (DDEConv > 0) {  // DDE conversation established.
   // Save an open file.
   DDEExecute(DDEConv, "Save c:\tfw\scripts\example.slt");
   // Save the active file.
   DDEExecute(DDEConv, "Save");

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

Command: Close
Description: This item closes a file in the Script Editor.
Params: The file to be closed. If no file is given, the active file is closed.
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("SEDITOR", "SEDITOR");
 if (DDEConv > 0) {  // DDE conversation established.
   // Close a specific file.
   DDEExecute(DDEConv, "Close c:\tfw\scripts\example.slt");
   // Close the active file.
   DDEExecute(DDEConv, "Close");

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

Command: Compile
Description: This item compiles a script.
Params: The file to be compiled. If no file is given, the active file is compiled. If the
                 given file is not currently loaded in the editor, it is opened first.
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("SEDITOR", "SEDITOR");
 if (DDEConv > 0) {  // DDE conversation established.
   DDEExecute(DDEConv, "Compile c:\tfw\scripts\example.slt");

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

Command: Exit
Description: This item causes the Script Editor to close.
Params: None.
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("SEDITOR", "SEDITOR");
 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: Activate
Description: This item makes the Script Editor the active applicaton.
Params: None.
Example:

```
main()
{
 int DDEConv;

 DDEConv = DDEInitiate("SEDITOR", "SEDITOR");
 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.");
 }
}
```

The <filename> Topic:

This topic is used to perform operations on a specific file, represented by <filename>. If <filename> is not open in the Script Editor, the DDE conversation will fail. This topic contains both DDE Execute and Request items.

Command: Save
Type: Execute
Description: Save the file to disk.
Params: None.
Example:

```
main()
{
 int DDEConv;
 str ExampleFile[] = "c:\tfw\scripts\example.slt",
     Buffer[255];

 DDEConv = DDEInitiate("SEDITOR", ExampleFile);

 // The following IF statement is used to ensure that the file
 // that we want to talk to is open.
 if (DDEConv == 0) {  // DDE conversation not established.
                      // Make sure file is open.
   // Open conversation with the editor
   DDEConv = DDEInitiate("SEDITOR", "SEDITOR");
   if (DDEConv == 0) {  // Can't find the editor
     prints("Could not connect to the Script Editor.");
     return 0;
   } else {
     // Open the file we want to work with
     Buffer = "Open ";
     StrCat(Buffer, ExampleFile);
     DDEExecute(DDEConv, Buffer);
     // Terminate the conversation with the editor.
     DDETerminate(DDEConv);
     // Try to establish conversation agin.
     DDEConv = DDEInitiate("SEDITOR", ExampleFile);
     // Fall through to the error checking below.
   }
 }

 if (DDEConv > 0) {  // DDE conversation established.
   DDEExecute(DDEConv, "Save");

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

Command: Close
Type: Execute
Description: Closes the file. Once file is closed, no further operations can be performed on it, so the conversation should be terminated.
Params: None.
Example:

```
  main()
  {
   int DDEConv;
   str ExampleFile[] = "c:\tfw\scripts\example.slt",
       Buffer[255];

   DDEConv = DDEInitiate("SEDITOR", ExampleFile);

   if (DDEConv == 0) {  // DDE conversation not established.
     // The file must not be opened.  No further processing needed.
     prints("File not open.");
   } else {  // DDE conversation established.
     DDEExecute(DDEConv, "Close");

     // close the DDE conversation.
     DDETerminate(DDEConv);
   }
  }
```

Command: Compile
Type: Execute
Description: This item compiles the file.
Params: None.
Example:

```
main()
{
 int DDEConv;
 str ExampleFile[] = "c:\tfw\scripts\example.slt",
     Buffer[255];

 DDEConv = DDEInitiate("SEDITOR", ExampleFile);

 // The following IF statement is used to ensure that the file
 // that we want to talk to is open.
 if (DDEConv == 0) {  // DDE conversation not established.
                      // Make sure file is open.
   // Open conversation with the editor
   DDEConv = DDEInitiate("SEDITOR", "SEDITOR");
   if (DDEConv == 0) {  // Can't find the editor
     prints("Could not connect to the Script Editor.");
     return 0;
   } else {
     // Open the file we want to work with
     Buffer = "Open ";
     StrCat(Buffer, ExampleFile);
     DDEExecute(DDEConv, Buffer);
     // Terminate the conversation with the editor.
     DDETerminate(DDEConv);
     // Try to establish conversation agin.
     DDEConv = DDEInitiate("SEDITOR", ExampleFile);
     // Fall through to the error checking below.
   }
 }

 if (DDEConv > 0) {  // DDE conversation established.
   DDEExecute(DDEConv, "Compile");

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

Command: Code
Type: Poke
Description: This item inserts text into the file at the current cursor position. If any text is highlighted, it will be replaced by the new text.
Params: The text to insert into the file.
Example:

```
main()
{
 int DDEConv;
 str ExampleFile[] = "c:\tfw\scripts\writer.slt",
     Buffer[255];

 DDEConv = DDEInitiate("SEDITOR", ExampleFile);

 // The following IF statement is used to ensure that the file
 // that we want to talk to is open.
 if (DDEConv == 0) {  // DDE conversation not established.
                      // Make sure file is open.
   // Open conversation with the editor
   DDEConv = DDEInitiate("SEDITOR", "SEDITOR");
   if (DDEConv == 0) {  // Can't find the editor
     prints("Could not connect to the Script Editor.");
     return 0;
   } else {
     // Open the file we want to work with
     Buffer = "Open ";
     StrCat(Buffer, ExampleFile);
     DDEExecute(DDEConv, Buffer);
     // Terminate the conversation with the editor.
     DDETerminate(DDEConv);
     // Try to establish conversation agin.
     DDEConv = DDEInitiate("SEDITOR", ExampleFile);
     // Fall through to the error checking below.
   }
 }

 if (DDEConv > 0) {  // DDE conversation established.
   DDEPoke(DDEConv, "Code", "main()^M");
   DDEPoke(DDEConv, "Code", " {^M");
   DDEPoke(DDEConv, "Code", "  prints(^"Hi, I'm a SALT script ");
   DDEPoke(DDEConv, "Code", "written by another script.^");^M");
   DDEPoke(DDEConv, "Code", " }^M");
   DDEExecute(DDEConv, "Save");
   if (DDEExecute(DDEConv, "Compile"))
     MsgBox("Script Writer", "Script written and compiled.", 0);

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

Command: Cursor
Type: Poke
Description: This item moves the cursor to a specified position in the file.
Params: A string containing the line and column to move to. If blank, the cursor is positioned at the beginning of the file. If -1, -1 is specified, the cursor is moved to the end of the file.
Example:

```
main()
{
 int DDEConv;
 str ExampleFile[] = "c:\tfw\scripts\example.slt",
     Buffer[255];

 DDEConv = DDEInitiate("SEDITOR", ExampleFile);

 // The following IF statement is used to ensure that the file
 // that we want to talk to is open.
 if (DDEConv == 0) {  // DDE conversation not established.
                      // Make sure file is open.
   // Open conversation with the editor
   DDEConv = DDEInitiate("SEDITOR", "SEDITOR");
   if (DDEConv == 0) {  // Can't find the editor
     prints("Could not connect to the Script Editor.");
     return 0;
   } else {
     // Open the file we want to work with
     Buffer = "Open ";
     StrCat(Buffer, ExampleFile);
     DDEExecute(DDEConv, Buffer);
     // Terminate the conversation with the editor.
     DDETerminate(DDEConv);
     // Try to establish conversation agin.
     DDEConv = DDEInitiate("SEDITOR", ExampleFile);
     // Fall through to the error checking below.
   }
 }

 if (DDEConv > 0) {  // DDE conversation established.
   // Move to Line 3, Column 1
   DDEPoke(DDEConv, "Cursor", "3,1");
   // Move to beginning of file
   DDEPoke(DDEConv, "Cursor", "");
   // Move to end of file
   DDEPoke(DDEConv, "Cursor", "-1,-1");

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

Command: Code
Type: Request
Description: This item returns the line of code at the current cursor position in the active file.
Example:

```
main()
{
 int DDEConv;
 str ExampleFile[] = "c:\tfw\scripts\example.slt",
     Buffer[255];

 DDEConv = DDEInitiate("SEDITOR", ExampleFile);

 // The following IF statement is used to ensure that the file
 // that we want to talk to is open.
 if (DDEConv == 0) {  // DDE conversation not established.
                      // Make sure file is open.
   // Open conversation with the editor
   DDEConv = DDEInitiate("SEDITOR", "SEDITOR");
   if (DDEConv == 0) {  // Can't find the editor
     prints("Could not connect to the Script Editor.");
     return 0;
   } else {
     // Open the file we want to work with
     Buffer = "Open ";
     StrCat(Buffer, ExampleFile);
     DDEExecute(DDEConv, Buffer);
     // Terminate the conversation with the editor.
     DDETerminate(DDEConv);
     // Try to establish conversation agin.
     DDEConv = DDEInitiate("SEDITOR", ExampleFile);
     // Fall through to the error checking below.
   }
 }

 if (DDEConv > 0) {  // DDE conversation established.
   // Print the first three lines of the file.
   DDEPoke(DDEConv, "Cursor", "1,1");  // Move to line 1
   DDERequest(DDEConv, "Code", Buffer);
   prints(Buffer);
   DDEPoke(DDEConv, "Cursor", "2,1");  // Move to line 2
   DDERequest(DDEConv, "Code", Buffer);
   prints(Buffer);
   DDEPoke(DDEConv, "Cursor", "3,1");  // Move to line 3
   DDERequest(DDEConv, "Code", Buffer);
   prints(Buffer);

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

Command: Cursor
Type: Request
Description: This item returns the cursor positions of the active file.
Example:

```
main()
{
 int DDEConv;
 str ExampleFile[] = "c:\tfw\scripts\example.slt",
     Buffer[255];

 DDEConv = DDEInitiate("SEDITOR", ExampleFile);

 // The following IF statement is used to ensure that the file
 // that we want to talk to is open.
 if (DDEConv == 0) {  // DDE conversation not established.
                      // Make sure file is open.
   // Open conversation with the editor
   DDEConv = DDEInitiate("SEDITOR", "SEDITOR");
   if (DDEConv == 0) {  // Can't find the editor
     prints("Could not connect to the Script Editor.");
     return 0;
   } else {
     // Open the file we want to work with
     Buffer = "Open ";
     StrCat(Buffer, ExampleFile);
     DDEExecute(DDEConv, Buffer);
     // Terminate the conversation with the editor.
     DDETerminate(DDEConv);
     // Try to establish conversation agin.
     DDEConv = DDEInitiate("SEDITOR", ExampleFile);
     // Fall through to the error checking below.
   }
 }

 if (DDEConv > 0) {  // DDE conversation established.
   DDERequest(DDEConv, "Cursor", Buffer);
   printsc("The cursor is at: ");
   prints(Buffer);

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