

RealThinClient SDK – Lesson 4 – Send large files out

This demo continues over Demo 3 (Sending small files). Now, we are going to make a webserver that is able to send files no matter their size.

We must be careful with our server's memory, we may have a really big file in our server and we don't want to use all of it.

At our first two examples we have been sending all of our content at once on the **OnCheckRequest** and **OnDataReceived** events of our **RtcDataProvider** components.

Now, to be able to send large content out (big files), additionally to the events that we have been using from our **RtcDataProvider** components, we will need the **OnDataSent** event. This event will be called after all of the data prepared with previous calls to the Write function have been sent out to the client and all sending buffers are empty. Also, we will need to do some checks on our **TRtcDataServer's** Response property.

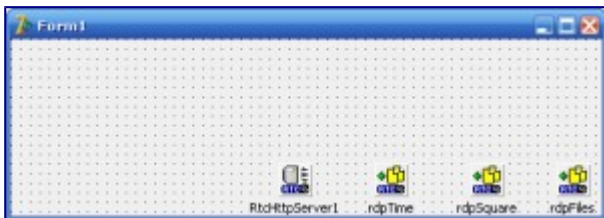
We will use as a base the code developed for Lesson 3 so now, we will be able to send large files out from our server. We are defining a memory limit of 16K for every single event call in order to keep our server's memory usage as low as 16K for any client requesting files from our server.

What are we going to do?

- Open our Lesson 3 code.
- Make change to the **OnCheckRequest** event for our **RtcDataProvider** component.
- Update our **OnDataReceived** event for our **RtcDataProvider** component.
- Implement our **OnDataReceived** event in **OnDataSent** event for our **RtcDataProvider** component.

Steps.

1. Open our Lesson's 3 code.



Project Opened

2. Change our RtcDataProvider's OnCheckRequest event.

We will change the OnCheckRequest event for our last **RtcDataProvider** component, to set the **Response.ContentLength** value to the file size. To do this, double-click on the last **RtcDataProvider** component to jump to the **OnCheckRequest** event provider. There is only a small addition to the event, but to keep it all in one place, here's the complete new event implementation:

```
1 procedure TForm1.rdpFilesCheckRequest(Sender: TRtcConnection);
2 var
3 vsFileName : string;
4 begin
5 with TRtcDataServer(Sender) do
6 begin
7 vsFileName := GetFullFileName(Request.FileName); //(1)
8 Accept;
9 Request.Info['FName'] := vsFileName;
10Response.ContentLength:=File_Size(vsFileName);//(2)
11WriteHeader;//(3)
12end;
13end;
```

What are we doing?

1. As we did in Demo 3, we will store the file name in our request so we don't have to recreate it again later.
2. We need to set the **Response.ContentLength**, to tell the **RtcDataServer** how large the content (data) in our response will be. If we do not set the **Response.ContentLength**, **RtcDataServer** will assume that the first event which calls **Write** has prepared the complete response and will calculate the **ContentLength** for us.
3. We will send the Response Header out, so we don't have to call **Write** in case our File size is zero.

3. Change our RtcDataProvider's OnDataReceived event.

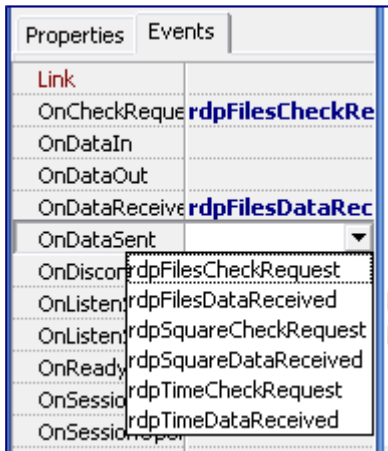
Now, we will update the **OnDataReceived** event of our **RtcDataProvider**, in order to send only limited amount of data out (we are setting that limit to 16KB).

What are we doing?

1. Check if we have to send more data.
2. Only continue if the file hasn't changed in size, if not then Disconnect
3. Calculate how much content of our file we still need to send
4. Limit the length to send at once to 16KB. Remember we don't want to use all of our server's memory.
5. Send **viSent** bytes from our file (**vsFileName**), starting at position **Response.ContentOut**.
6. Disconnect the client, because our file has changed and we have sent the wrong header and file beginning out.
7. If the File isn't found then we send an error message to the client.

4. Implement OnDataReceived event in OnDataSent event.

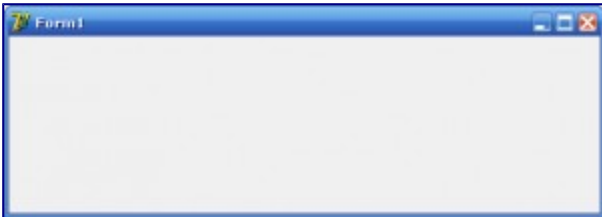
Now, we must implement the **OnDataReceived** event for the **OnDataSent** event. If we don't do this, then our server will just send the Header and the first 16K of Data and the transference will never end.



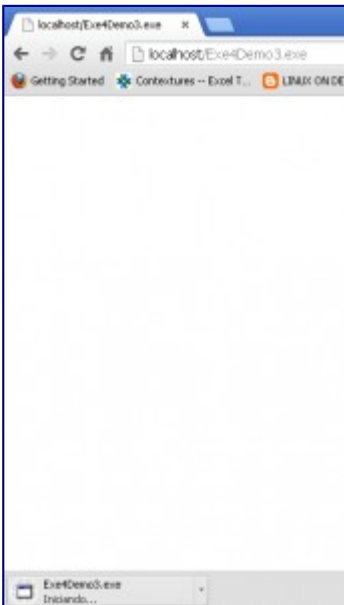
OnDataSent Event

5. Compile and run our Project.

Now we can compile and run our project. Remember that you can put any big file in the **data** subdirectory and it will be served using just 16K and not the whole server's memory (if the file is really big of course).



Server Running



File being received in browser