FOUNTAINHEAD

NEW YORK
2ND MAR
2016

Baruch
COLLEGE

ADVANCED EXCEL: RTDs

~ Excel RTDs ~

Excel RTD

Excel RTD was introduced with Excel 2002. It is a mechanism for getting real-time data into Excel in an efficient and high capacity way.

Push/Pull Update Mechanism

Excel uses a hybrid push-pull mechanism that works in the following way.

When the RTD server has new data, rather than forcing the data on Excel, it merely notifies Excel that data is ready. It does this by calling the Excel callback function `UpdateNotify()`.

When Excel is ready (which is usually every couple of seconds or so), it will request the data from the RTD server and pull it into Excel and display it. It does this by calling the RTD callback function `RefreshData()`.

This hybrid push-pull mechanism ensures that Excel remains responsive and is not “hammered” by incoming data. Because of this it is a great way to delegate computational tasks to run in the background. This way Excel remains usable and, when the data is ready, it simply pops up in Excel and can be used like any other data element in a worksheet cell.

RTD() Function

RTD servers are exposed in Excel using the Excel worksheet function `RTD()`. The `RTD()` function has this form:

```
=RTD(RTDProgID,ServerName,topic1,[topic2],...,[topic28])
```

where:

- `RTDProgID` is the program ID of the real-time data (RTD) server to which to connect.
- `ServerName` is the name of the remote machine on which to run the RTD server. This is only applicable to out-of-process (EXE) RTD servers.
- `topic1,[topic2],...,[topic28]` are topics, which are simple text strings, passed to the RTD server. These topics request a specific item of real-time data from the RTD server. Each unique set of topics is treated as a single data item request and is assigned a unique topic ID by Excel.

It is also possible to wrap the `RTD()` function using VBA to make user friendly functions. For example, the RTD function call used in our example, which looks like this in raw form:

```
=RTD(“RTDDemo”,,”Fibonacci”,A2)
```

Note: A2 is a cell reference to a cell containing a number, say, 17.

This could instead be wrapped in a VBA helper function to look like this in a workbook cell:

```
=Fibonacci(A2)
```

Parameter Passing with RTD

The RTD function does not have parameters, per se, but “topics”. Topics are simply strings, but in the `RTD()` function you can use cell references and numbers and they will be converted to strings automatically by Excel before being passed to `RTD()`.

It is also the case that values returned by `RTD()` are treated like any other cell value in Excel. Values returned by `RTD()` can be used to drive other Excel formulas, so in that sense the RTD integrates very well and is just like any other Excel formula.

Passing anything other than simple values to and `RTD()` can be a bit of a challenge, as it simple takes and returns strings. But there are tricks to get around many of these problems. For example, an array of data can be returned as a string such as this:

```
“{1, 2, 3, 4; 10, 20, 30, 40; 100, 200, 300, 400}”
```

And using Excel’s `Evaluate()` function it can be made into a cell range that looks like this:

1	2	3	4
10	20	30	40
100	200	300	400

Note: There is a 256 character limit on the length of the string passed to `Evaluate()`.

~ RTD API ~

RTD API Functions

An RTD server is simply a COM object that implements the `IRtdServer` interface, which requires the following methods to be implemented:

- `ServerStart()`
- `ServerTerminate()`
- `ConnectData()`
- `DisconnectData()`
- `Heartbeat()`

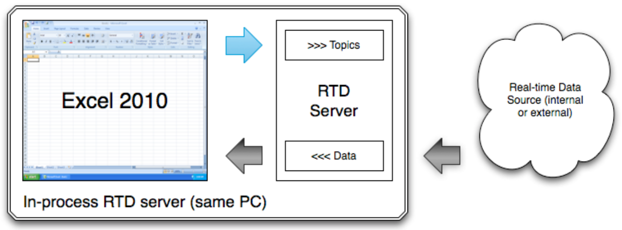
The RTD server communicates with Excel via a callback object that implements the `IRTDUpdateEvent` interface and which is passed to the RTD as a parameter to the `ServerStart()` method.

© 2012-2015, Andrew Sheppard, all rights reserved.

~ In-Process RTD ~

In-Process RTD Server

Excel RTD servers can be in-process (DLL) or out-of-process (EXE). In-process RTD servers run in the same memory space as Excel.


Excel 2010
>>> Topics
RTD Server
<<< Data
Real-time Data Source (internal or external)
In-process RTD server (same PC)

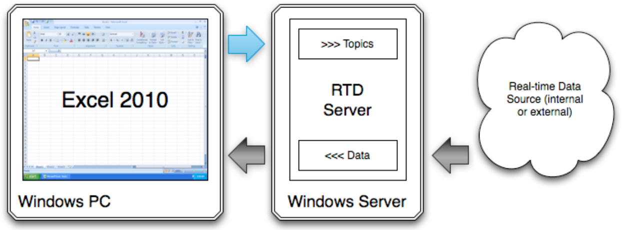
In-Process RTD Server

~ Out-Of-Process RTD ~

Out-of-Process RTD Server

An out-of-process RTD server runs outside Excel's memory space. It runs as a separate process either on the local machine, or on a remote machine (DCOM).

This gives a lot of flexibility in terms of adding supercomputing capability to Excel because you can shift the computational burden of your model outside Excel to where you have better computational resources. Either locally, such as onto the GPU, or remotely onto a cluster, or even into the Cloud.

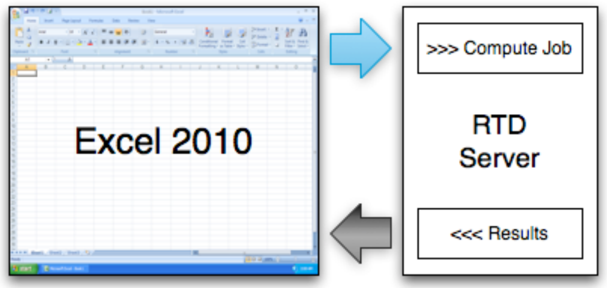


Out-of-Process RTD Server

~ RTD Compute ~

RTD Compute Engine

RTD was originally developed to feed real-time data into Excel. But there is nothing stopping you from using it to drive a calculation engine that then delivers the results of computationally intensive calculations as items of real-time data.



Using an RTD Server as a Calculation Engine

So how would this work?

First you would enter a function call using the RTD() function:

```
=RTD("RTDDemo",,"Fibonacci",
10)
```

The `RTD()` function would return immediately letting the user know that a computation has been dispatched:

	A	B	C
1	Calculating …		
2			
3			

In the meantime, the user can continue working normally in Excel. No "hourglass", no interruption. This is one of the major advantages of RTD.

Then, when the calculation is complete, and Excel is ready to receive the result, the placeholder value of "Calculating …" will be replaced with the result:

	A	B	C
1	34		
2			
3			

~ GPU RTD ~

RTD Supercomputer

Of course, you can make a RTD calculation engine work with any sort of computational resource, either locally or remotely. This could mean CPU, GPU, even FPGA; or clusters using any of these technologies. And since RTD can work over the network, it could potentially make use of compute resources in the Cloud, or Cloud+GPU.

RTD GPU Supercomputer

By wrapping GPU kernel programs using C# to make them easily callable from .NET, it then becomes quite easy to call GPU functions from a cell in Excel that offloads the calculation to the GPU.

~ Pros/Cons for RTD Supercomputing ~

Pros

These are the pros of using RTD:

- Long running calculations are dispatched in the background leaving Excel usable. No "hourglass".
- Offloads the computational burden from Excel. Local or remote.
- Looks and behaves like a native Excel function. Well integrated.

Cons

Here are the cons of using RTD as a supercomputing add-on for Excel:

- Passing parameters to, and returning results from, RTD() can pose some difficulties. But there are workarounds.

© 2012-2015, Andrew Sheppard, all rights reserved.