Document Id	TN1047
Last Modified Date	10/14/2015
Legacy Tech Note #	2

SUMMARY

What is Multiplexing Overload?

The practical limit for the number of conversations* that can be serviced by an I/O Server using DDE is about a hundred. Beyond that, the I/O Server will update slowly and become unresponsive. Even before your computer begins to run out of system resources, the I/O Server will first run out of DDE bandwidth on the server. This is called "Multiplexing Overload."

Multiplexing Overload applies to all Wonderware® and third party I/O Servers, regardless of the type of network communications that it uses (for example, Data Highway Plus, Modbus, and so on).

Note The term "conversation" is defined here as one DDE conversation between two DDE-aware applications. For example, if you have two
active DDE Access Names inside your InTouch application that have an Application Name of "ABKF2," but they each have a different Topic Name,
then there would be two conversations. As you add more Topic Names, the number of conversations will increase.

Misconceptions on I/O Server's Multiplexing Capabilities

Many people believe that a I/O Server's multiplexing capabilities are unlimited. Also, many people believe that they can designate an unlimited number of DDE clients to point to the I/O Server in their InTouch application with no adverse effects. This is not true.

An I/O Server can service as many DDE clients as the machine's resources will allow. But, there is a practical limit to the number of DDE clients that an I/O Server can service effectively.

SITUATION

Example InTouch Configuration

Relying on these common misconceptions, many people, unfortunately, create a system similar to this: an InTouch application is created that gathers data from about five PLCs. Each PLC contains data that can be updated at different rates. Therefore, the developer creates three topics for each PLC: Fast, Medium and Slow. However, with just one copy of WindowViewer running, the I/O Server needs to service 15 active conversations. (Three topics per PLC multiplied by five PLCs equals 15 conversations.)

When the application is almost finished, the developer then decides to connect another ten computers onto the network that will only run WindowViewer. NetDDE is started on each computer and the developer points the ten new WindowViewer (VIEW) nodes to the I/O Server that is running on the first node. Each new VIEW node has the same number of topics as the first node (that is, Fast, Medium and Slow) and each VIEW node looks at the same number of PLCs (five).

Because Wonderware I/O Servers are intelligent, the server will only poll once for each item, then multiplex the resulting data back to the DDE clients that are interested in the data. 8/22/13 Avoiding I/O Server Multiplexing Overloads https://wdnresource.wonderware.com/support/kbcd/html/1/avoiding.htm 2/5 Therefore, even if there are 11 DDE clients that are interested in a single point on one PLC, the server will poll it only once and then send the data back to each of the 11 clients.

Using this example system, here is the resulting number of topics that the I/O Server must multiplex the data back to the clients:

> **11** VIEW nodes (the DDE clients)
>
> **FIVE** PLCs that are read by each node
>
> **X** **THREE** topics per PLC
>
> _____
>
> **165 TOTAL ACTIVE CONVERSATIONS**

That is, 11 VIEW nodes * five PLCs * three topics per PLC = 165 active conversations.

If the developer adds another PLC to the network, here is how many active conversations there will be:

> **11** VIEW nodes (the DDE clients)
>
> **SIX** PLCs that are read by each node
>
> **X** **THREE** topics per PLC
>
> _____
>
> **198 TOTAL ACTIVE CONVERSATIONS**

This is almost a 16% increase in the number of conversations!

If the developer then adds another VIEW node to the network, here is how many conversations there will be:

12 VIEW nodes (the DDE clients)

SIX PLCs that are read by each node

X   **THREE** topics per PLC

_____

216 TOTAL ACTIVE CONVERSATIONS

This is another 8% increase in the number of conversations, and a 24% increase from the original network configuration, just by adding only one PLC and one VIEW node!

ACTION

How to Avoid Multiplexing Overloads

So how do you keep the number of DDE conversations between your client nodes and the I/O Server from overloading the server? Answer: You make the WindowViewer node (where the I/O Server is running) the "true" I/O Server. You do this by creating two InTouch applications: the Master Data Collection Application and the Client Application.

Application #1: Master Data Collection Application ("Master" Application)

This InTouch application has no windows, scripts, logging or alarms. The computer does not even need a keyboard, mouse or monitor. It only needs the following:

- Microsoft Windows NT4.0 (SP5)
- the NETDDE services (Network DDE, Network DDE DSDM, ClipBook Server) must be started
- the executables for the I/O Server and VIEW.EXE should be in your Windows NT Startup program group.

Application #1 resides on the node with the I/O Server. (Many people prefer developing this application first and refer to it as the "master" application.) The purpose of the master application is to service the needs of the other nodes on the network.

You only create one tag for each data point that is needed by any PLC on the network. This way, only one VIEW node looks at the I/O Server (and therefore at the PLCs) instead of all of the VIEW nodes. The tags will have DDE items that actually refer to the data points

within the PLCs. (Note that you can still have hot backup PLCs. See Creating Hot-Backup PLCs later in this topic for more information.)

The master application should also have the DDE Access Names point to the I/O Server that is running locally. You should still have three topics (that is, Fast, Medium and Slow) for each PLC and you should still spread your data points across the three topics.

Application #2: "Client Application"

This InTouch application has all the necessary windows, scripts, loggings and alarms. It runs on the nodes where the I/O Server is not running.

Application #2 (or the "client" application) has tags whose Item names are the same as the tagname. However, this application only has one DDE Access Name. This Access Name should have the Application "\\MASTER\VIEW" and the Topic "Tagname".

When the client application starts up, it will create only one active conversation between itself and the master application (Application #1).

Conversations are Decreased

Using the concept of the master and client applications (Applications #1 and #2), what is the resulting number of active conversations?

Here is the number of conversations between the Master node and the I/O Server:

> **ONE** VIEW node
>
> **SIX** PLCs that are read by each node
>
> **X**   **THREE** topics per PLC
> _____
>
> **18 TOTAL ACTIVE CONVERSATIONS**

That is, one node * six PLCs * three topics per PLC equals only 18 active conversations (and not 165 conversations like in our original example!).

If you add another PLC to the network, here is how many conversations there will be:

> **ONE** VIEW node
>
> **SEVEN** PLCs that are read by each node

X   **THREE** topics per PLC

---

21 TOTAL ACTIVE CONVERSATIONS

This is an increase of only three conversations (and not 33 conversations as in our original example).

What happens if you add another ten VIEW nodes? Nothing. The number of conversations between the Master node and the I/O Server does not increase if you add more VIEW nodes that point to the Master node.

Then, what is the number of conversations between the Master node and the remaining ten VIEW nodes?

TEN VIEW node

X   **ONE** topic per PLC

---

10 TOTAL ACTIVE CONVERSATIONS

Here is what happens if you add another VIEW node:

11 VIEW node

X   **ONE** topic per PLC

---

11 TOTAL ACTIVE CONVERSATIONS

This is an increase of only one conversation (and not 18 conversations as in our original example!).

Here is what happens if you double the number of VIEW nodes to 22:

22 VIEW node

X   **ONE** topic per PLC

---

22 TOTAL ACTIVE CONVERSATIONS

Again, this is an increase of only 11 conversations.

Note on Scaling DDE Tagnames for the Master and Client Applications

InTouch will automatically scale the "raw" tagname values to the Engineering Units. That is, the tagname values displayed in InTouch will be adjusted based on the values set for Min/Max Raw and Min/Max EU. (No scaling will take place if the Min/Max Raw values are equal to the Min/Max EU values.)

When the raw tagname values are received by the master application, it provides the scaled data (that is, the EU values) to the requesting client applications. However, in order to display the EU values from the master application "as is" without further scaling, remember to set the Min/Max Raw values equal to the Min/Max EU values in each client application.

For more information on scaling DDE tagnames, see "Scaling DDE Tagnames" in the InTouch User's Guide.

Creating Hot-Backup PLCs

Using the solution of creating a "master" InTouch application on the node with the I/O Server (the master node), the number of conversations is greatly reduced. But, how do you create hot-backup PLCs? You add another standby master node. This node is identical to the primary master node except for its node name. In Application #2, which is the client application, you create a Condition script that monitors the value of the DDEStatus bit which monitors the DDE conversation between WindowViewer and the master node. The body of the Condition script will have one line:

SetDdeAppTopic("MyOneAccessName", "\\MASTER2\VIEW", "");

"MyOneAccessName" is the DDE Access Name and "\\MASTER2\VIEW" is the Application Name (where MASTER2 is the node name of the standby master node). If the DDEStatus bit goes to a zero (which means the conversation with the primary master node was terminated), then the Condition script will automatically switch your InTouch application

to MASTER2. You can optionally add a button to one of your InTouch application windows that allows the user to switch back to the primary master node once he believes the conversation has been reactivated.

Conclusion

When you create an InTouch application that uses NetDDE, you should expect that at some point in time two, three or more WindowViewer nodes will be added to your system. (Once your users see how easy it is to add more InTouch nodes to the network, they will add more and more nodes as time goes by.) Eventually, your I/O Server will begin to suffer from Multiplexing Overload and the application that you originally created will update more and more slowly. This is caused by the dramatically increasing number of DDE conversations that are created as each new VIEW node is added.

You can easily avoid this situation by creating a "master" InTouch application on the node that is running the I/O server and a "client" InTouch application on the remaining nodes. This way, only one VIEW node is interfacing with the I/O Server to receive the data updates instead of all of your VIEW nodes.

ATTACHMENTS AUTHOR NOTES

Created: August-14-1995
Updated: September 22, 2000
Author: S. Weinrich