ARCHITECTURE GUIDE

Description

This guide will walk though typical Architectures for implementing Wonderware InTouch in a Terminal Services Environment. For a complete deployment guide of InTouch for Terminal Services please reference the published documentation here.

<table>
<thead>
<tr>
<th>Author</th>
<th>Michael Walker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Publish Date</td>
<td></td>
</tr>
<tr>
<td>Applies to Software</td>
<td>InTouch Classic, InTouch for System Platform</td>
</tr>
</tbody>
</table>
Concepts and Terminology

This section will outline some of the basic terminology and concepts around the InTouch for Terminal Services architecture.

**Terminal Services**

Terminal Services is a windows component that provides the technologies that enable users to access Windows-based programs installed on a Windows Server. This term is still widely used, but has been replaced by the Remote Desktop Services Role in Windows Server 2008. Essentially the term "Terminal Services" applies to the windows component installed on Windows Server 2003 and earlier Windows operating systems. Windows Server 2008 uses the term "Remote Desktop Services" in place of Terminal Services.

**Remote Desktop Services (Role)**

Remote Desktop Services (formerly Terminal Services), is a server role in Windows Server® 2008 R2. This role provides technologies that enable users to access Windows-based programs installed on a Remote Desktop Session Host (RD Session Host) server, or to access the full Windows desktop.

**Remote Desktop Services (Role Services)**

Several role services can be enable when using the Remote Desktop Services Role. It is important to understand what functionality these roles services provide within Remote Desktop Services.

**Remote Desktop Gateway** enables authorized users to connect to virtual desktops, Remote-App programs,
session-based desktops over a private network or the Internet.

**Remote Desktop Connection Broker** allows users to reconnect to their existing virtual desktop, RemoteApp programs, and session-based desktops. It enables even load distribution across RD Session Host servers in a session collection or across pooled virtual desktops in a pooled virtual desktop collection, and provides access to virtual desktops in a virtual desktop collection.

**Remote Desktop Session Host** enables a server to host RemoteApp programs as session-based desktops. Users can connect to RD Session Host servers in a session collection to run programs, save files, and use resources on those servers. Users can access Remote Desktop Session Host server by using the Remote Desktop Connection client or by using RemoteApp programs.

**Remote Desktop Virtualization Host** enables users to connect to virtual desktops by using RemoteApp and Desktop Connection.

**Remote Desktop Web Access** enables users to access RemoteApp and Desktop Connection through the Start Menu or through a web browser. RemoteApp and Desktop Connection provides users with a customized view of RemoteApp programs, session-based desktops, and virtual desktops.

**Remote Desktop Licensing** enables a server to manage Remote Desktop Services client access licenses (RDS CALs) that are required for each device or user to connect to a Remote Desktop Session Host server. RDS CALs are managed using the Remote Desktop Licensing Manager application.

**Remote Desktop Connection (Remote Desktop Protocol)**

Remote desktop connection is the client application used to remotely access a Windows server running Remote Desktop Services. When connecting it uses the Remote Desktop Protocol to present a user with the "session" on the server. Several software companies have built technologies that either leverage RDP or deploy similar technologies that implement with Remote Desktop Services. These companies include Wyse, ACP and Citrix to name a few. These companies are mainly focused on the connection to and management of a Remote Desktop Services environment. Another term noteworthy term is "Desktop Virtualization" where software is used to separate the Windows desktop environment and associated application software (such as Wonderware InTouch) and the physical device that is used to access it.

**Wonderware InTouch for Terminal Services**

InTouch for Terminal Services is a variation of the regular InTouch and is intended for use on a server with Remote Dekstop Services (Formerly Terminal Services) enabled. InTouch is installed on the RDS server and will allow multiple clients to connect and gain the functionality of a running InTouch application, without the additional software / hardware requirements on each client computer. In most cases, the clients hardware requirements are low while the RDS server's hardware requirements are high due to the resource usage needed for providing application functionality to multiple clients. The below figure represents multiple devices connecting to a RDS server running an InTouch Application. Note that additional software / hardware may be needed to run a RDS session on a device that does not contain the Remote Desktop Client, for example Apples iPad."
The diagram below represents a typical Remote Desktop Services (Terminal Services) environment using Windows Server 2008 R2. This includes the use of the RDS and the Remote Desktop Gateway for connecting external clients.

The diagram below represents multiple RDS servers within a NLB cluster, this allows multiple clients to connect to a "pool" (essentially a group) of RDS servers, based on the clients requirements, user profile, etc... the connection can be launched on a specific RDS server. Software like ACP ThinManager has built in load balancing capabilities that can be leveraged rather than using the Remote Session Broker. Also note that the Remote Session Broker doesn't have be installed on a stand-alone server as seen below, and can exist on a RDS server within the cluster.
Notes and Warnings

Some final notes and warnings on the Architecture of the InTouch for Terminal Services Environment. Review these before designing the Architecture to ensure a good design as well as alleviate any "Gotchas" during implementation.

Architecture is driven by the Application!

Several factors needed to be taken into consideration when designing the InTouch for Terminal Services Environment, such as the following:

**RDS Server Resource Requirements:**

Review the current environment to gain an understanding on the resource requirements for each application. For example, if an InTouch application running on 20 client machines uses around 1 GB of memory during normal operation, then the Remote Desktop Server at a minimum needs a 20 GB of memory to be able to serve these clients.
sessions for each of the 20 clients. Also, additional memory will be needed for the overhead of the operating system hosting the RDS server. The same type of review needs to be completed for CPU and Network Bandwidth usage.

**Redundancy / Failover Application Requirements:**

In some cases, the application may require some form of failover or redundancy in the event of RDS server failure. Typical architecture includes "Pairs" of terminal servers, which are load balanced at 50% so in the event one of the servers in the pair goes down, the application(s) that were hosted on the failed RDS server will now run on single RDS server which will be at 100% resource utilization during the failure.

**Security Requirements for each application:**

Typically a domain controller is used to manage user profiles across multiple Remote Desktop Servers, in the event that a domain controller is not available, use local users on each of the Remote Desktop Servers.

**Connectivity to other devices / systems:**

Depending on how the application is integrated with other systems, the Remote Desktop server will need access to PLC's, Databases, File Shares, etc.. just as if they were running on a stand alone client. It is important to review the application and ensure that the necessary connectivity to other systems is available.

**Application Development may be Required!**

When an application is moved from a stand-alone architecture to a Remote Desktop Services (Terminal Services) architecture, some changes may need to be made. These changes typically are the following:

**Access Name Configuration:**

Access Names that connect intouch to PLC's and other field devices may need to be updated to reflect how the RDS server will connect to these devices.
**Alarm Configuration:**

In a change from Windows Server 2003, Windows Server 2008 no longer supports the /console switch as a means of starting the remote desktop (RDP) client, also known as Session 0 or Terminal Server Console session. In Windows Server 2008, Session 0 is no longer an interactive session, and is reserved only for Windows services. Windows Server 2008 treats all remote connections as remote RDP sessions regardless of /console, /admin, or any other switches used to make the connection. This impacts InTouch HMI functionality such as Alarm Manager that depends on the Terminal Server Console session. InTouch uses SuiteLink to communicate with the alarm sub system. The SuiteLink handle is based on the IP address. Considering this, alarm queries configured to point to specific client IP addresses using `\TSENodename\ClientIPaddress\InTouch\$System` syntax will behave as follows:

If running multiple sessions on the same client, the alarm query will always return the first View session's alarms. Therefore, do not run multiple sessions on the same client when using this query syntax. In addition, a user who logs on to a session via one client, then disconnects from that session (without shutting down View) and reconnects to the same session from another client will result in the alarm query returning alarms from the initial session. The same results will occur if one disconnects from a session without shutting View down and then logs on to the same client with another user. Therefore, make certain to shut View down properly, prior to disconnecting from a session.

Use the following syntax to query the *local* session’s alarms:

- `\InTouch\$System`

Use the following syntax to query the *Terminal Server console’s* alarms:

- `\TSENodename\InTouch\$System`

Using the above query syntax, running multiple sessions on one client can be done without encountering the problems mentioned above.

**Application Synchronization via a TagServer Application:**

https://knowledge.insourcess.com/Wonderware_InTouch_HMI/Architecture_Articles/Intouch_for_Terminal_Services_Preparati…

Updated: Sat, 14 Dec 2019 10:34:02 GMT

Powered by
A common requirement of customers using InTouch for Terminal Services is to have multiple operators running the same InTouch application. The nature of Terminal Server is that each session is independent of the other sessions and also the Console. This means that if each session opens the same window of the same InTouch application and changes the value of a slider linked to a memory tag, then each session will see a different value for the same memory tag. In other words, each session has its own instance of the same memory tag. Most Terminal Server applications do not want independence of memory tag values, but they need to have the memory tags centralized and also centralized alarms and history. The centralization of memory tags is accomplished by using the Tag Server architecture. Centralization of alarms and history is accomplished by configuration of the alarms and history so that only the Console application writes to history files and writes to the alarm database. Note that no centralization of IO tags is necessary in the InTouch application, because the source of the IO tags is the IO Servers. The Tag Server architecture has been used in InTouch applications for years. More information about this architecture can be found in the InTouch User's Guide. The implementation of Tag Server in a Terminal Server contains a Tag Server application running on the Terminal Server Console and InTouch client applications running in the Terminal Server sessions that have remote tags referencing tags in the Tag Server application on the Console.