CA Client Automation

Data Transport Service Administration Guide

12.9

CA Technologies Product References

This documentation set references to the following CA products:

- CA Advantage® Data Transport® (CA Data Transport)
- CA Asset Intelligence
- CA Asset Portfolio Management (CA APM)
- CA Business Intelligence
- CA Common Services™
- CA Desktop Migration Manager (CA DMM)
- CA Embedded Entitlements Manager (CA EEM)
- CA Mobile Device Management (CA MDM)
- CA Network and Systems Management (CA NSM)
- CA Patch Manager
- CA Process Automation
- CA Service Desk Manager
- CA WorldView™

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 1: Welcome to Data Transport Service

Data Transport Service (DTS) is a comprehensive transport service network that lets you transfer data across platforms and transport protocols. Using DTS, you can perform the following list of operations:

- Create, schedule, and initiate data transfers
- Monitor and control active data transfers
- Create logical groupings of machines on the network to manage data transfers from multiple end systems
- Optimize network performance by establishing transmission settings to control how machines send and receive data transfers
- Remotely administer the DTS network

Note: The term Windows refers to the Microsoft Windows operating system. For the detailed list of supported platforms refer to Compatibility Matrix.

This section contains the following topics:

Benefits and Features

DTS is supplied with and used by CA client solutions, such as Client Automation. These client solutions install DTS automatically with a predetermined configuration, or they may let you customize your configuration of DTS during the installation process. The features of DTS are enhanced when you install the CA Common Services (CCS) components, WorldView and Event Management.

For the instructions on using a CA solution specifically designed to work with DTS, see the solution's documentation and online help. Generally, the solution's online help describes how to use the solution after you install and configure DTS on all desired machines in your network. Even if DTS is installed automatically when you install the solution, you may need to configure DTS on some machines in your network. For example, to create links between machines, configure the protocols used by a machine or set other machine-related properties.

Note: For more information, see the Configuration Policy section of the *DSM Explorer Help*.

The benefits and features of DTS are grouped into the following categories:

Network Administration

Data Transport Service can be used to schedule, initiate, monitor, and control data transfers without any network administration being performed. However, if needed, network administrators can perform network administration tasks, such as establishing customized network topologies and optimizing network performance.

Integration with CA Common Services

Data Transport Service provides more functionality by integrating with the WorldView and Event Management components of CCS.

Note: WorldView is only supported on Windows with an MS-SQL database. For more information about supported databases and operating environments, see the *Implementation Guide*.

Data Transfer

Data Transport Service enables the administrator to create and initiate data transfers across various platforms and transfer protocols. Data transfers can be activated immediately or scheduled for a later date and time.

The benefits and features of DTS are available through interfaces:

- Graphical user interface for network administration (Administration Client Interface)
- Graphical user interface for configuring policies for DTS agents in Client Automation
- Client application program interface (API)
- Command-line interface

Network Administration

Network administrators perform network administration tasks, like establishing customized network topologies and optimizing network performance. You can use DTS to schedule, initiate, monitor, and control data transfers with or without any network administration.

Note: Network administrators, who install and maintain DTS network components or install and maintain the CA client solution that includes DTS, should be familiar or possess a general knowledge of CCS and the operating system conventions of the computers on which you are installing and administering agents.

Administrators can use the results of the DTS registration process, which locates all DTS manager and agent components on the network, to establish a particular network topology. You can use this topology to create customized transfer routes through the network.

DTS administrators can also establish transmission settings to control how various DTS machines send and receive data transfers. You can use these transmission settings to optimize network performance by establishing the maximum size of parcels to transfer, the time to wait between parcel sends, and the maximum number of concurrent processes permitted.

Also, administrators can remotely configure the runtime parameters for *legacy* Data Transport Service r1, r2, and r 3 agents and manager components (NOS, TOS, and SOS). You can configure all of the administrative features previously described using CA Common Services. These features are also configurable through the DTS client API.

In Client Automation, DTS agent and manager components are controlled by configuration policies that are set up using the DSM Explorer.

More information:

Viewing Configuration Policies (see page 49)
Modifying Configuration Policies (see page 50)

Command Line Interface

DTS provides a command line interface for network administrators to schedule and initiate transfers, and to start, stop, and display the status of DTS components. CA solutions that work with DTS provide an additional command line interface, the Data Transport Service Command Line Interface (DTSCLI), for initiating, monitoring, or controlling data transfers from the command line.

More information:

Transfer Object Server (TOS) (see page 26)

dtscli Command

The dtscli command lets you do the following:

- Create transfers, transfer jobs, and schedules
- Delete transfers, transfer jobs, and schedules
- Retrieve the status of transfers, transfer jobs, and schedules
- Add, insert, and remove transfers from transfer jobs
- Activate, suspend, resume, abort, and reset transfer jobs
- Adjust the priority of a transfer job
- Add, insert, and remove transfer jobs from schedules
- Enable and disable schedules
- Perform agent-to-agent transfers
- Perform managed transfers

The dtscli command includes functions like HTTP transfers (Internet downloads), creating and manipulating transfer jobs and schedules, and accepting input from a command file.

Note: In Client Automation, dtacli is superseded by dtscli. Although dtacli is still provided, we recommend that you migrate to dtscli.

More information:

Creating Transfer Objects (see page 47)
DTS Security (see page 34)
dtacli and dtscli Commands (see page 91)

Self Discovery

DTS automatically adds the DTS agents and managers to the Management Database (MDB) as they send a notification message to the TOS when they start. This process is called *Self Discovery*. By default, all agents and managers are configured to notify the TOS when they start, for example, after a restart or after a new computer starts for the first time. Also, agents notify the TOS when their addresses change, which is most common with roaming agents, like laptop computers or agents that use a DHCP address. The TOS passes the message to the NOS.

When the NOS receives a notification message from an agent or manager (either directly or forwarded by the TOS), the NOS checks the MDB to see if that agent or manager is already in the repository. If not, the NOS adds the agent or manager to the repository.

Self Discovery is disabled by default in Client Automation. To enable this process, change the value of the Self discovery configuration policy in the Network Object Server (NOS) policy group.

More information:

Modifying Configuration Policies (see page 50)

Dynamic Containers and Dynamic Routing

Dynamic containers let machines route in a dynamic environment. Dynamic containers are Data Transport classes that you create and manage in WorldView, as part of the MDB. When you create a dynamic container, you specify a range of addresses or a subnet. All agents whose IP addresses are in this range become members of the dynamic container. By constructing links between machines and dynamic containers, it is then possible to define paths between machines based on the machine's IP address. When a machine's IP address changes such that it becomes a member of a different dynamic container, then the path to that machine also changes.

Note: DTS WorldView dynamic container membership only supports IPv4 address ranges.

Error Messages

Whenever a managed DTS transfer fails, the error message is passed to the TOS. Applications using DTS can query the TOS for the error message and display it to their users. External filters can provide detailed error information, which aids in the identification and correction of problems.

Support for Transferring Large Files

DTS does not have file size restrictions, but the file system of the operating system may have such restrictions. Most legacy operating systems imposed a limit of approximately 2 GB on the largest file that their file systems could store. More recent operating systems have overcome this limitation, and DTS can transfer files or directories larger than 4 GB on such systems.

Note: For more information about file system specifications for large file support, refer to the documentation for your operating system.

PIF Installation Packages for UNIX

DTS UNIX installation programs use the product interchange format (PIF) format, the CA standard for packaging and installing software solutions on UNIX systems. Developers create PIF installation packages using a CA utility called the Software Manager Installer for UNIX, hereafter called the PIF Installer. The PIF Installer is also provided as part of the CA solution's PIF installation package. When you run the PIF installation package, the PIF Installer lets you install and remove the solution easily and efficiently on the target UNIX computers.

Note: For more information, see the *PIF Packager and Installer Administration Guide*.

DTS and CA Common Services Integration

DTS does not require CA Common Services (CCS); however, without this solution, DTS is limited to functions that do not require or use CCS. DTS provides additional functionality by integrating with the WorldView component of CCS, which is optionally installed when you install a CA client solution that includes DTS.

The existing installations of a CCS component satisfy these requirements. For best results, we recommend that you install the most recent release of any CCS component that you use, or intend to use, from the CA Common Services files that are supplied by the installation program of your CA client solution.

Note: You cannot use the features of CCS if you do not install CA Common Services on at least one computer in your DTS network.

Important! When upgrading CCS by installing either Unicenter DSM or CA Client Automation Release 12.9, and if discovery has previously been done under CCS r3.0 using short names (for example, executing the dscvone -n command), subsequent discovery must be carried out with the 'remove suffix' option enabled. Unless this is done, previously discovered objects and any DTS link that is established to them in the 2D map is lost. The reason is that when the CCS component is upgraded to Release 12.9, it will automatically re-discover the network and create TCP/IP objects with Fully Qualified Domain Names in place of the original ones.

Ensure that you enable 'remove suffix' option for dscvrbe (classic discovery) after Unicenter DSM or Client Automation is installed, using the -3 option, as follows:

dscvrbe -3 <*suffix to remove*>

Example:

dscvrbe -3 ca.com

Note: The 'remove suffix' option does not exist for Continuous Discovery in CCS 11.2.

Note: For detailed information about all upgrades, see the "Upgrading and Migration Considerations" chapter in the *Implementation Guide*.

WorldView

If you install CA Common Services on your network, DTS is closely integrated with it. CA Common Services provides WorldView, Enterprise Discovery, Business Process Views™, 2D Map, ObjectView, MDB, and other components, which in turn provide DTS with important features like routing, containers, dynamic containers, links, throttling, and several protocols for data transfers.

You can administer DTS using WorldView and the following features that are installed with DTS:

- Several Business Process Views in WorldView
- Context menus to include DTS-specific options
- Several DTS server-specific forms (usually through context menus)

WorldView provides the following features, all of which DTS can use to view and manage your network resources:

Business Process Views

During the DTS Auto Discovery process, a DTS Business Process View is automatically created in the WorldView Managed Objects folder, which contains the discovered DTS managers and agents. The DTS administrators can use this Business Process View to establish a DTS network topology. In that topology, you can configure customized transfer routes. The DTS Business Process View can be used to simplify the maintenance of a large network of machines that are involved in DTS activities.

Real World Interface

The Real World Interface lets management applications display 2D graphical representations of the resources they manage and the relationships between those resources.

2D Maps

The two-dimensional animation, used by the Real World Interface, provides administrators with a realistic view of their enterprise. This view helps administrators to more easily resolve problems by letting them travel through the enterprise and monitor the status of their resources. Also, the 2D Maps come with built-in geographic maps, through which you can view managed resources by location.

The DTS administrators use the 2D Maps to display and monitor their DTS resources and to establish their network topology.

Enterprise Discovery

The WorldView Enterprise Discovery process detects or discovers network entities and resources and then populates the MDB with objects representing those entities and their relationships. The Real World Interface displays these objects and the entities they represent, monitor, and control.

DTS Interfaces to WorldView

The DTS Admin Client dialogs are accessible through context menus for objects in WorldView, and DTS Auto Discovery is available from the context menus of the DTS Business Process Views.

Installing and Upgrading Business Process Views

Typically, when you install Data Transport Service on an existing computer running Client Automation, the installation process automatically executes the dtsbpv program. The dtsbpv program automatically decides whether the DTS Business Process Views you must install, upgrade, or both install and upgrade. The program then automatically performs the required installation, upgrade, or both of the Business Process Views on that computer.

In other words, the dtsbpv program adds to or upgrades the Data Transport Service classes and objects that are stored in the MDB. Normally, the installation or upgrade of these Business Process Views occurs automatically, without user input, during the Data Transport Service installation process.

Post-Installation Tasks

On rare occasions, however, the dtsbpv program does not run successfully during the Data Transport Service installation.

If you are not certain whether the dtsbpv program completed successfully or if you know that the program failed, run it manually. Also, even if the initial installation of the Business Process Views was successful, you can reinstall them again later, if necessary. For example, you can reinstall the Business Process Views to re-initialize the Data Transport Service NOS classes.

Note: Running the dtsbpv program item clears any previously established Data Transport Service classes and objects and replaces them with an empty Data Transport Service framework.

Note: Run the dtsbpv program only on the computer on which Client Automation and the Data Transport Service Network Object Server (NOS) component have been installed.

Running dtsbpv from the Command Line

To run the dtsbpv program from the command line, enter the following command:

```
dtsbpv
```

Uninstalling and Reinstalling Business Process Views

If you want to uninstall (dtsbpv -u) and reinstall your Data Transport Service Business Process Views on a computer, enter the following commands from the command line:

```
dtsbpv -u
dtsbpv
```

More information:

Business Process Views Installation Fails (see page 88)

Enterprise Management

DTS uses many Enterprise Management functions, including Event Management and CCS calendars.

Event Management

Event Management provides for automated handling of messages and events. The Event Management console provides a window into the Enterprise Management GUI that monitors system events as they occur. DTS events, like starting and stopping the DTS servers, requesting a transfer to a machine object that cannot be located, DTS failures, and security validation failures, are also monitored. Any messages the DTS events generate are automatically sent to the Event Management console where they are identified by a specific prefix.

Depending on your sites requirements, you can configure Event Management to process event messages on individual servers or to redirect them to a central server or other Event Management servers. You can collect related messages network-wide and can display them on a single console or can send them to multiple locations as needed.

Through Event Management, you can identify specific messages important to your operation, and then specify the action to perform when they occur. You can enhance the messages to track source, or to customize them to your sites needs.

Note: In Client Automation, DTS uses Event Management but not directly. DTS is integrated with the Common Application Framework (CAF) to which events are passed, and CAF passes the events to Event Management.

CCS Calendars

The DTS implementation of Event Management includes CCS calendars, which the Enterprise Management component supplies. CCS calendars let the SOS schedule recurring transfers. Without calendars, DTS could schedule only one-time (single-occurrence) transfers.

Calendars provide a way to create schedules containing time and date information by which you can initiate various jobs or functions. DTS uses calendars to create complex schedules for data transfers. You can use these schedules to initiate transfers at a future date or time, to create regularly scheduled transfers, and to prevent transfers from occurring at a particular time.

Note: You can access scheduling features through the client API.

Data Transfers

Using the features of DTS, you can create and initiate data transfers across various platforms and transfer protocols. You can activate these transfers immediately or schedule them to occur later.

You can monitor active data transfers to view their status. Among the details you can monitor are: the percentage of the transfer that is completed and the total number of parcels transferred. Also, you can suspend, abort, or resume active data transfers, if they were previously suspended.

For DTS, the data transfer features discussed here are available to you through the client API. DTS serves as the underlying service of CA solutions, such as Client Automation. These applications provide access to the data transfer functions through their own GUIs for transferring data.

Consult your CA account representative for information about CA applications that provide a GUI to DTS.

Alternative Routing

If a transfer succeeds through the most efficient route, then no alternative route is tried. However, if the most efficient route fails, the alternative routes are tried for a pre-specified number of times, or until the transfer completes or fails.

The DTS network administrators assign efficiency of each possible route based on their observation of network performance.

Discreet Transfers

Discreet transfers are transfers that are sent or received in the background when the sending and receiving computers are not heavily loaded. From the load, the DTS agent determines when to send the transfer and calculates the optimal transfer rate, so the discreet transfer has a minimal impact on the computers performance and the users productivity.

The rate at which a discreet transfer progresses is determined automatically and dynamically by the DTS agent. The DTS agent monitors the load (both CPU usage and the machine's network usage) on the sending and receiving computers. When a computer is loaded lightly, the DTS agent uses the spare resources. However, if the load increases, the agent relinquishes resources, giving priority to other tasks the computer is performing. This way, discreet mode optimizes the computer’s performance while performing transfers.

Maximum Number of Transfers per Job for Fanout, Broadcast, or Multicast

You can specify the maximum number of transfers permitted in each transfer job generated when you activate a fanout, broadcast, or multicast. If the fanout, broadcast, or multicast generates more transfers than this maximum value, DTS automatically creates as many additional transfer jobs as necessary to hold all the transfers generated by the fanout, broadcast, or multicast.

Note: For detailed information, see the *DTSCLI Command Reference Guide*.

Staging Directory Configuration

DTS automatically cleans, or deletes, the staging files of successful transfers. For failed and aborted transfers, DTS uses the StagingFileLifetime parameter to determine if and when to delete the staging files of such transfers.

Note: For detailed information, see the *Data Transport Service WorldView Administration Client Help*.

Auditing

DTS administrators can customize audit messages for common events, like the failure of a DTS agent or TOS or an aborted transfer job. Any message that you customize replaces the corresponding default message that DTS would issue under the same conditions to the Client Automation common event component.

For example, before customization, DTS sends a default "Transfer Object Server failure" message to the TOS log file and to Client Automation if TOS fails during normal operation. However, if you customize the TOS_FAIL= *audit token*, then DTS sends your customized message instead of the default message to the Client Automation.

Customized messages provide administrators with increased flexibility for monitoring and handling errors. For example, if a transfer fails or is suspended, the TOS can be configured to output the source, destination, and the text of your customized message.

Note: In Client Automation, communications, auditing, tracing, event management, compression, encryption, are handled by common components in the Common Application Framework (CAF). For an overview, see the *Implementation Guide*.

More information:

Customizing Audit Messages (see page 56)
Changing Audit Levels (see page 53)

Handling of DHCP and Modem Connections

Notification parameters are provided for DTS agents that connect to their TOS using dial-up machines or DHCP. The notification parameters let these agents, which can be connected to the network infrequently or for short time periods, select for outstanding transfers as soon as they connect to a TOS.

HTTP Protocol Support

The HTTP protocol is the most common protocol for transferring data from World Wide Web servers to browsers. A DTS user can create, maintain, and perform transfers to receive data from an HTTP server or its proxy server, even if the server does not have a DTS agent installed. DTS thus supports downloading files from the Internet.

Note: For more information, see the *Data Transport Service WorldView Administration Client Help*.

Protocol Selection

You can configure the DTS agents to load only the protocols you specify upon startup instead of all protocols DTS supports. This results the slave agents to load the protocols that perform the transfer.

External Protocols

You can use external protocols that are installed on the selected computer but not supplied by DTS. You can write protocol libraries for unsupported protocols.

Chapter 2: Architecture

This chapter presents an overview of the architecture of Data Transport Service, including brief descriptions of its various components and some features.

This section contains the following topics:

Overview

DTS is a flexible service in terms of how to configure the different components in your network. Before you configure your DTS network, understand the role that each component plays. The central component of DTS, the Data Transport manager, operates as a distributed server. Applications connect to the Data Transfer manager to perform or schedule data transfers. Any computer that involves the transfer of actual data must have the DTS agent installed.

In Client Automation, communications, auditing, tracing, event management, compression, encryption, are handled by common components in the Common Application Framework (CAF). For an overview, see the *Implementation Guide*.

Note: The previous releases for DTS that are upgraded to Unicenter DSM r11.1 do not integrate with CA Common Services. This means that any DTS advanced network configuration set up with a previous version of CA Common Services are not available in r11.1. If, however, a previous release is upgraded to Unicenter or Client Automation Release 12.9, this does not apply, that is, the existing network configuration is available.

Note: Installing Client Automation r12 on top of CCS r3.0 is successful; however, any DTS data that has been configured in CCS will no longer be available.

DTS Manager

The DTS network environment comprises multiple domains, each consisting of a DTS manager. The DTS manager consists of the following manager components, each of which plays a unique role:

- Network Object Server (NOS)
- Transfer Object Server (TOS)
- Schedule Object Server (SOS)

Note: The Transfer Object Server (TOS), Network Object Server (NOS), and Schedule Object Server are commonly referred to as object servers, DTS servers, and servers throughout the documentation. However, in formal DSM terminology, they are DTS managers.

Network Object Server

The Network Object Server (NOS) manages the DTS machine and machine group information that is stored in WorldView™, part of the MDB. You can define complex routing topologies through the network using the properties of information that is stored in objects. Also, the TOS uses the NOS to establish routing topologies and optimal transfer paths throughout the DTS network.

You do not generally manipulate objects that are stored in the MDB through the client API. Typically, you use WorldView to configure and manage DTS network objects.

The DTS uses four-phase optimization to ensure your data transfers in the most efficient manner. The NOS is responsible for carrying out Phases 1 and 2 of the optimization process, which is Network Route Analysis and Transfer Property Resolution, respectively.

More information:

Phase 1: Network Route Analysis (see page 61)
Phase 2: Transfer Property Resolution (see page 62)

NOS Classes

The classes that the NOS maintains are the following:

Machine Class (DTMachine)

A machine object represents any computer that is part of a DTS domain, or, any machine where you have installed a DTS server, agent, or both. Machine objects define properties associated with the actual computer, like name and description, and properties defining transmission settings to use during data transfer. If a computer has more than one communications interface, the machine object may reference one or more interface objects. DTMachine is a virtual, extended superclass with real subclasses of host, workstation, and unclassified TCP. So, when using DTMachine, you do not need to know whether a specific computer is classified as a host, workstation, or unclassified TCP object.

The MDB does not provide or contain the DTMachine class. The NOS provides the DTMachine. DTMachine exists to ease the use of objects in the MDB that represent computers, and to extend those objects such that they contain many DTS service-specific properties. The properties of machine objects affect how a data transfer is performed, and you can modify these properties to meet your needs. For example, you can modify the appropriate property to limit the rate of data transfers out of a machine, specify the protocols available on the machine, or indicate that a machine is available for receiving a broadcast data transfer.

Machine Group Class (DTMachineGroup)

A machine group object consists of a logical grouping of computers, other machine group objects, or both. You can establish these objects. Machine group objects let users and administrators create their own unique, logical, view of a DTS domain. Machine group objects are independent of the defined network topology and do not affect transfer routes through the network.

Machine Container Class (DTContainer)

A machine container object defines a common set of transfer properties for a number of machine objects. These properties include the protocol and other communication settings the computers use during a data transfer. Administrators can use machine container objects to establish properties for a group of machine objects, by placing the machine objects into a container object and then setting the properties of the machine container object.

Dynamic Container Class (DTDynamicContainer)

The Dynamic container class (DTDynamicContainer) functions much like the DTContainer class but has an additional property: a range of addresses or a subnet that defines the members of the container. All agents whose IP address is in this range become members of the dynamic container.

Dynamic containers let users create routes between machines and a dynamic container. A dynamic container stores objects whose addresses change over time, for example, machines that use DHCP or dial-up connections. When you view DTS objects in WorldView, if the DTS object you select is a dynamic container, then Class field shows DTDynamicContainer and a second tab (Membership) appears next to the Object Details tab. That tab lets you view and modify the individual addresses and address ranges belonging to the dynamic container.

Note: DTS WorldView dynamic container membership only supports IPv4 address ranges.

Link Class (DTLink)

A link object defines a DTS communications link between two different machine or machine container objects. DTS administrators define these links that identify source and destination computers, with a common set of transfer properties associated with the link. Links define fixed data transfer routes through the enterprise.

Interface Class (DTInterface)

An interface object defines a physical interface associated with a particular computer. You can associate specific protocol characteristics with the interface. By assigning a priority property to the interface, users can specify the order for evaluating the interfaces when determining the route and properties to use for a transfer. Interfaces associated with the responding computer are sequentially compared to those of the initiating computer until an interface with matching characteristics is found.

Transfer Object Server (TOS)

The Transfer Object Server (TOS) is responsible for initiating, controlling, and monitoring all user requested data transfers, and accepting and processing requests from the client API. When the TOS activates a data transfer, it sends a message to the initiating DTS agent, telling it to transfer the data. The TOS is then responsible for processing progress and status messages that are returned from the initiating agent.

The TOS optimizes those requests by making the most efficient use of network resources, it manages hops (intermediate nodes), fanouts, and broadcasts. The TOS uses any available network topology information that it obtains from the NOS, and instructs the DTS agent on the sending machine to perform the data transfer. The TOS also lets applications monitor and control the transfers that are submitted.

You can connect to the TOS through the client API and can create filter objects, transfer objects, and can transfer group objects. You set object properties to define your data transfer requirements. You can then call the transfer group objects ACTIVATE method to perform the transfers you have defined. The TOS stores all these objects persistently in its own local object repository that is named the Transfer Object Repository (TOR).

Note: It is possible to perform agent-to-agent transfers without the TOS, using the DTS command-line interface.

The TOS directly accesses the MDB from its TOR and uses NOS to get information from WorldView, also part of the MDB. It uses the network object properties that are returned from the NOS to make decisions about the best way to transfer your data.

DTS uses the four-phase optimization for transferring your data in the most efficient way. The TOS is responsible for carrying out Phases 3 and 4 of the optimization process, Duplicate Transfer Resolution, and Transfer Mechanism Selection, respectively.

More information:

Phase 3: Duplicate Transfer Resolution (see page 64)
Command Line Interface (see page 11)
Phase 4: Transfer Mechanism Selection (see page 65)

TOS Classes

The classes that the TOS maintains are the following:

Transfer Class (DTTransfer)

A transfer object defines how to perform a data transfer. It encapsulates all the information required to carry out the transport of the specified data from one location to another. It also provides properties and methods that let the user monitor and control the data transfer. Transfer objects contain references to filter objects and machines. When a transfer object is created, the TOS uses the properties from the default transfer object for the new transfer. This lets you set the minimum amount of properties that allow the transfer to run. These properties are input (source data), output (destination data), initiator (source machine), and responder (destination machine).

Transfer Group Class (DTTransferGroup)

A transfer group object consists of a user-defined group of transfer objects. You can either activate these transfers sequentially or as a group (which takes full advantage of the activation algorithms in the TOS). The transfer objects cannot be activated, aborted, suspended, resumed, or reset individually. These methods can be performed on transfer group objects only.

Filter Class (DTFilter)

Filter objects define some operation to perform on the data to be transferred. This operation works either on the file (file based) or on buffers of data (parcel based) when the transfer is activated. Filter objects are associated with a transfer object and represent the actual filters or filtering that is applied to the data before, during, or after the data is either read or written, or both. These filters can include text reading, data encryption, and data compression. The properties of the filter objects determine the type of filtering used.

Schedule Object Server (SOS)

The Schedule Object Server (SOS) is used to establish and control scheduling requests that determine when transfer occurs. This schedule information is maintained in the form of schedule objects that are held in the Schedule Object Server's local repository. The Schedule Object Server monitors these objects and instructs the Transfer Object Server to activate the transfer group(s) referenced by the schedule objects when the specified date and time expire. You can access scheduling features through the client API.

Note: In Client Automation, the SOS is disabled by default. To enable it, enter the caf enable dtssos command from a command-line window or from the Windows Start menu's Run option. For more information, type <command> /? at the command prompt.

DTS Agent

The DTS agent is responsible for carrying out the physical transfer of data, which it does at the request of the TOS. The DTS agent supports multiple communication protocols, including TCP/IP, UDP/IP, PPP, SNA, SPX, Broadcast, and Multicast, using the CLI.

Note: DTS does not support the latest version of PPP communication protocol, PPPv6.

During any given data transfer, an agent takes on two personality characteristics which define its role (and responsibilities) during the transfer cycle. These characteristics dictate that the agent acts as either an initiator or a responder, and as either a sender or a receiver.

Initiator

If the agent is an initiator, the TOS directly instructs the agent to carry out the data transfer. The initiator agent is responsible for initiating the connection to the other agent (the responder) and for informing the TOS of transfer progress and status changes.

Responder

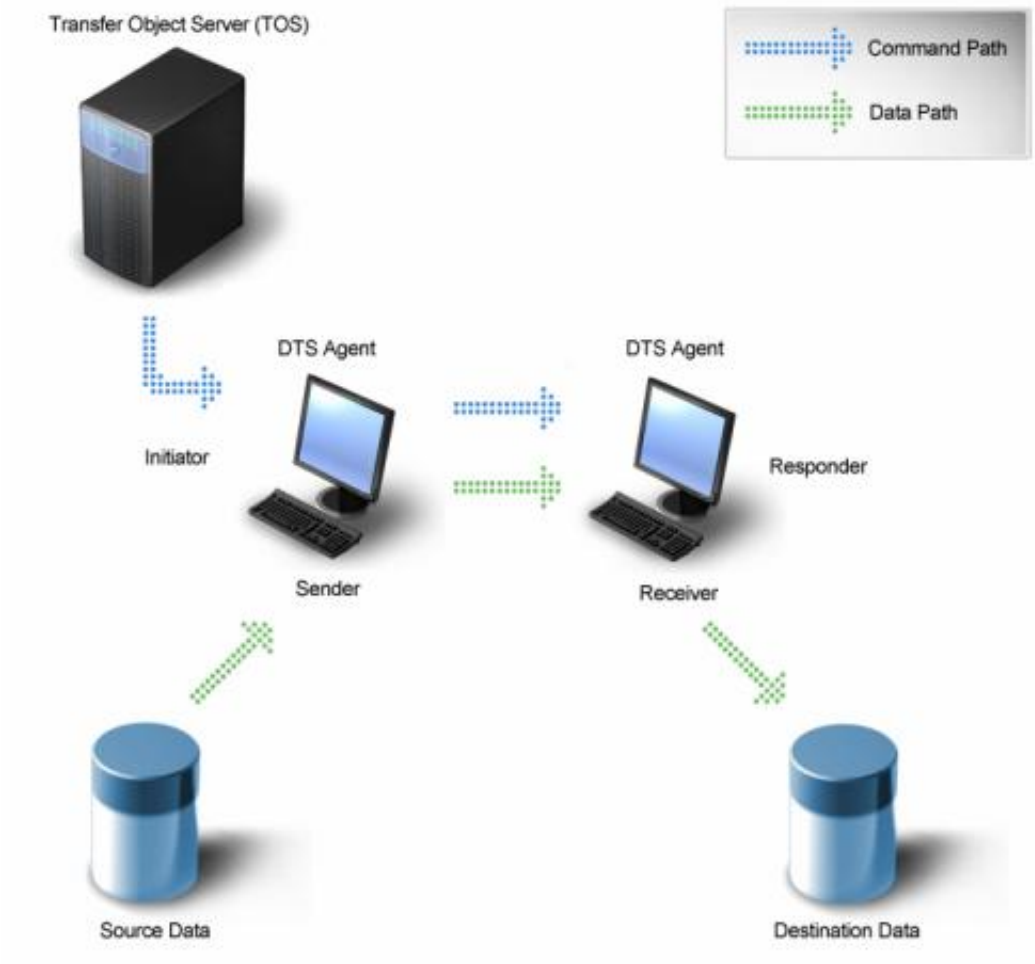
If the agent is a responder, then an initiator agent (and not the TOS) has connected to it and has told it what to do.

Sender and Receiver Tasks

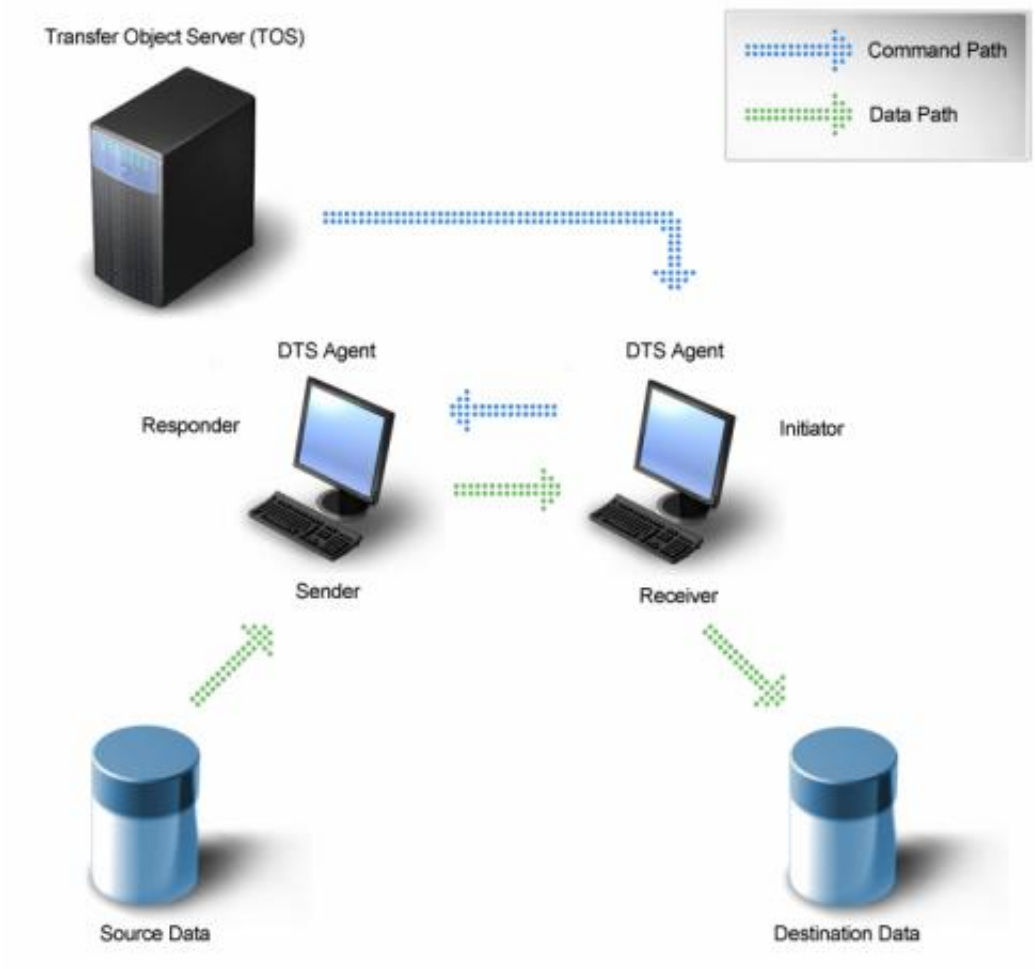
For the actual physical task of transferring data, the agent's job is straightforward. As a sender, the agent must read and process the source data and then send it to the receiver agent. As a receiver, the agent must receive data from the sender agent and then process and must write the data out to the destination location.

Note: An initiator agent is usually a sender and a responder agent is usually a receiver, however, it can be the reverse.

A typical case where the initiator agent is the sender and the responder agent is the receiver is illustrated in the following diagram:



The case where the initiator agent is the receiver while the responder agent is the sender is illustrated here:



Filters

The filter mechanism lets the agent read, process, and write data. Filters for files and parcels can read textual data and binary data, respectively, compress data, encrypt data, write binary data, decompress data, unencrypt data.

Agents read and send data from one to another in chunks that are known as parcels. A parcel is the amount of raw data a sender agent reads before it applies any parcel filters and before it sends the data across the network to the receiver agent. The default parcel size is approximately 500 KB. So, for example, the process of transferring 10 MB of data involves the reading and sending of 20 parcels.

You can create your own filters and corresponding filter objects for use during transfers, or you can use several predefined file and parcel filters.

Valid DTSCLI file filters include the following types:

- Directory filters
- Encryption filters
- File attribute filters

When adding the file filters, parcel filters can also be needed. To transfer a directory, file filters must be added to the dtscli command. If you try to transfer a directory without adding the directory filter, you get the following error message:

```
"Failed to open input data <c:\dtstemp> error=<Permission denied>".
```

Valid DTSCLI parcel filters include the following types:

- Binary (default) or text
- Compress
- Encrypt

The two main parcel filters that are used are either binary or text. Use the text filter if transferring text files between two different types of computers (Windows and UNIX, for example). This ensures that the text format is preserved; otherwise, always use binary filters.

Note: Binary and text filters are mutually exclusive: A transfer can either be text or binary but not both.

Filters usually work in pairs. The write filter must reverse an action that is performed by the read filter. Filters are applied in the following order:

- File Read—Works on a file when reading the source data
- Parcel Read—Works on a buffer of data when reading the source data
- Parcel Write—Works on a buffer of data when writing the destination data
- File Write—Works on a file when writing the destination data

Read the filters of the same type (file or parcel) are applied in the same order that they are specified on the command line. Write the filters of the same type are applied in the reverse order that they are specified on the command line.

DTS also supports external file and parcel filters.

Example: Transfer a Directory

- The following example transfers a directory using a short format for the directory filter:

```
dtscli -agent ipath=Gold::c:\dtstemp
    "rpath=Silver::c:\dtstemp"
    "f_filters=dir"
```

- This example performs the same operation but uses a long format:

```
dtscli -agent "ipath=Gold::c:\dtstemp"
    "rpath=Silver::c:\dtstemp"
    "f_filters=DIRTREE_READ:DIRTREE_WRITE"
    "p_filters=binary"
```

Example: Transfer a File as Text

- The following example transfers the file from Gold to Silver using the text filter and a short format:

```
dtscli -agent "ipath=Gold::c:\dtstemp\file.src"
    "rpath=Silver::c:\dtstemp\file.dest"
    "p_filters=text"
```

- This example performs the same operation but uses a long format:

```
dtscli -agent "ipath=Gold::c:\dtstemp\file.src"
    "rpath=Silver::c:\dtstemp\file.dest"
    "p_filters=TEXT_READ:TEXT_WRITE"
```

  Note: Only use text filters for transferring text files.

Note: For detailed information, see the *DTSCLI Command Reference Guide*.

More information:

Support for External Filters (see page 95)
Creating Transfer Objects (see page 47)

DTS Client API

The DTS client API consists of a library of functions that provide a programming interface to the various DTS functions. These functions include data transfer and scheduling, error logging and recovery, and network configuration information and monitoring facilities.

The client API is situated at the user application level and communicates with the DTS servers through a communications link, using a protocol such as TCP/IP. When the client API receives application requests, it forwards those requests to the DTS servers, and returns responses to the calling application.

Using the client API, applications can perform the following:

- Connect and disconnect from object servers
- Create and delete objects (instantiate and destroy)
- View and configure objects (get and set properties)
- Call functions (methods) associated with the objects
- Query the object repositories for objects that match certain selection criteria

The client API also provides a number of administration functions for viewing and updating both manager and agent run-time parameters.

More information:

Creating Transfer Objects (see page 47)

DTS Browser

If you have installed a CA solution that provides a transfer GUI for DTS, the browser lets you use that transfer GUI to create transfers by browsing and selecting drives, directories, and files on remote computers, instead of manually entering path names on the command line or in a file. For more information about how to use the features that the browser provides, see online help for the transfer GUI of the CA solution that you use to supplement DTS.

Note: In all earlier releases of DTS, the browser was named the Data Object Agent. The AS/400, OS/2, and OpenVMS components still use the term Data Object Agent rather than Browser or Browser Agent.

DTS Security

Security exists at several levels in the DTS architecture:

Data Transport Manager (DTS Manager)

You can activate security for each of the DTS manager components (TOS, NOS, and SOS) and for the DTS agent.

When you activate security (the security mode parameter for the computer is set to "fail"), agents connecting to a server or agent must supply a valid user and password of an existing user that is registered on the computer you are connecting to.

Data Transport Agent (DTS Agent)

When you activate security for an agent, any data transfers involving the agent must contain valid user and password combinations for the initiator and the responder computers.

Data

Encryption provides data security. The data is encrypted before transmission and unencrypted when it reaches its destination. You can use either parcel or file encryption filters.

More information:

dtscli Command (see page 12)

Chapter 3: Implementation

The information details about setting up a centralized DTS network. The actual implementation is performed by Client Automation, with DTS agent and manager components that are installed automatically in the appropriate places.

This section contains the following topics:

Typical Configuration

In a typical configuration, you can install the DTS manager components—the TOS, NOS, and SOS—on a Windows Server machine with a DTS agent. All machines that involve data transfer activities must have a DTS agent installed.

All DTS functions, which include transfer schedules, resolution of transfer routes, and initiation of the actual data transfers, are performed on the same machine. A single TOS, NOS, and SOS can perform these functions for your entire enterprise.

This type of setup allows for the most centralized way of controlling the DTS network.

The following models can aid in determining how the DTS architecture can best be implemented to suit your needs:

- Implementation model
- Object model

Other Possible Configurations

You can install the manager components on different machines in your network if you do not require all of the capabilities of the manager components on a single machine, or if circumstances are such that it is more beneficial for you not to have them on a single machine.

For example, if your site plans to initiate a heavy load of transfers, you can install the TOS on a separate machine other than the NOS and SOS to ease the load imposed on a single machine. If necessary, all three servers can be installed on separate machines.

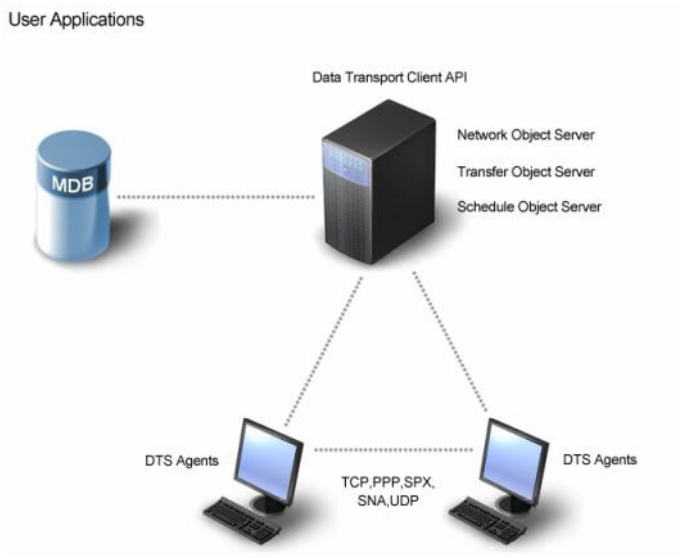
Note: The configuration options that are actually offered may also depend on the CA client solution that installs DTS.

Implementation Model

The implementation model shows how the product is implemented and how the service is provided on a high level:

- The client API tells the DTS manager what to do.
- The manager works out how to do it and then tells the DTS agent what to do.
- The agent does what it has been told and then tells the manager that the task is finished.
- The manager tells the client API that the task is completed.

The following diagram illustrates how the DTS architecture implements both client/server and manager/agent methodologies. Together, these methodologies ensure that the implementation possibilities are flexible and scalable. In this illustration, the manager doubles as the server end of the client/server solution:



Applications that want to use DTS call the functions that are contained in the DTS client API, which instructs the DTS manager to carry out the requested task. The DTS manager (DTM) includes three separate servers (NOS, TOS, and SOS). Each type of server has a specific role to play no matter how many instances are implemented.

To understand how the DTS components interact to perform a data transfer, consider the following diagram:

If a user has scheduled a transfer for a particular date and time. When that date and time arrive, the SOS recognizes that a transfer must occur, and tells the TOS to carry out the transfer. The TOS, upon receiving this request, instructs the initiating agent to transfer the data.

To achieve the levels of functionality, management, control, and flexibility that a mission-critical, enterprise-wide data transport solution demands, DTS uses a well structured object model. By creating instances of objects, setting properties, and calling methods, you can take full advantage of the powerful features of DTS.

Object Model

The object model explains how DTS is object-oriented (OO). Object-oriented means that to use the services available, you create objects (instantiate), configure them (set properties), and call functions (methods) associated with those objects. The DTS object model consists of a set of object classes that reference and interact with one another. To use DTS, ensure that you can do the following:

- Connect to an object server
- Create (instantiate) objects of the appropriate classes
- Set the properties of the objects to requisite values
- Call object methods

A brief description of each of the classes that DTS supports follows:

DTContainer

Overrides DTS properties for a list of DTMachine objects. You can treat several DTMachine objects as a single entity using DTContainers, when it comes to DTS network topology administration and configuration. This object contains a full set of DTS properties and references to one or more DTMachine objects.

DTDynamicContainer

Contains the same properties as the DTContainer class but also has an additional property: a range of addresses or a subnet that defines the members of the container. All agents whose IP address is in this range become members of the dynamic container. Dynamic containers let a user create routes between machines and a dynamic container.

DTFilter

Specifies how to read or write data, and any processing to perform on the data before or after the transfer. For instance, you can use filters like binary read, compression, decompression, binary write, and so forth.

DTInterface

Represents a communications interface on a computer.

DTLink

Represents a direct DTS communications link between one DTMachine (or DTContainer) and another. The link object creates specific routes through the DTS enterprise.

Each object server (Network, Transfer and Schedule) maintains objects of particular classes specific to the function of the server. Local object repositories (databases) store these objects persistently; one repository exists for each object server.

DTMachine

Defines a computer in the enterprise. If the computer has multiple communications interfaces fitted and available for DTS to use, then this object references one or more DTInterface objects.

DTMachineGroup

Represents a logical grouping of computers. This object references DTMachine objects or other DTMachineGroup objects.

DTSchedule

Specifies when a transfer should occur. This object references one or more DTTransferGroup objects. This object may also reference one or more calendars.

DTTransfer

Defines a single data transfer from one computer to another. Among other things, this object contains references to DTFilter objects.

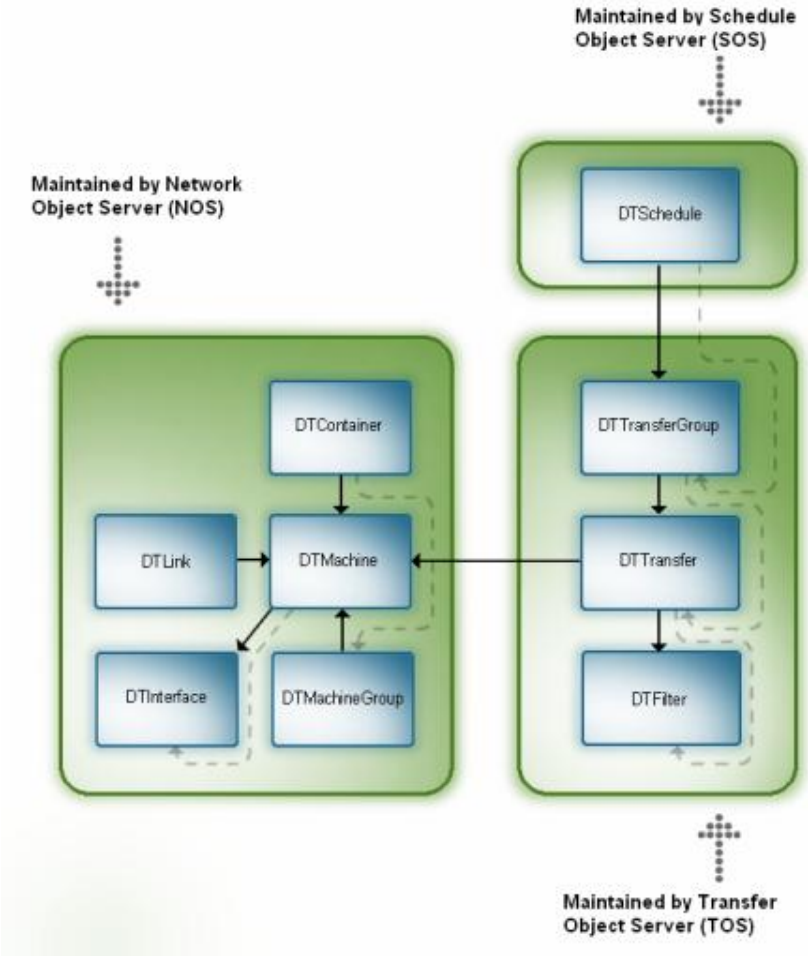
DTTransferGroup

Defines a group of transfers that are to be controlled together. It references one or more DTTransfer objects.

Note: The NOS uses the MDB as its persistent object store. DTS utilizes the network information already stored in the MDB.

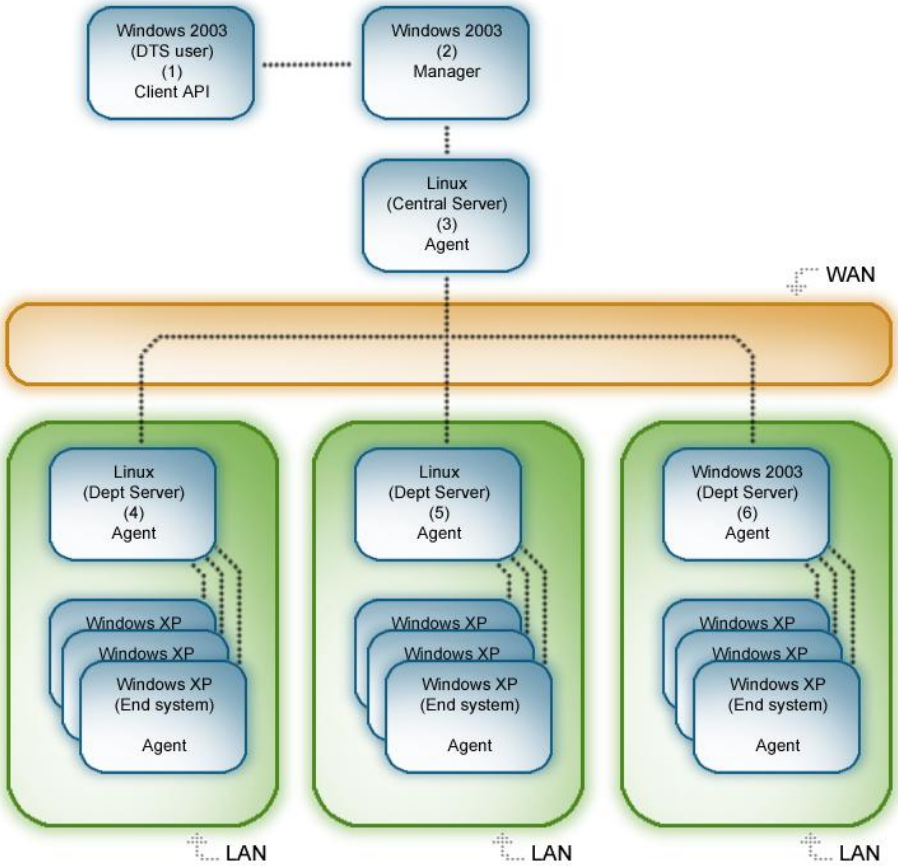
Referential Links Between Various Objects

The following diagram illustrates the referential links between the various object classes and which object classes are maintained by each of the servers:



DTS Scenario

To know how you initiate a data transfer, how you use the object model, and the role each of the individual DTS components plays, a typical DTS scenario follows:



If a DTS administrator wants to perform a set of data transfers at 10:00 p.m. each week night. A large binary data file residing on a Linux server machine must be transferred to numerous Windows XP computers, which are spread across three separate local area networks (LANs). Each LAN has a departmental file server connected to it. The central server connects to the LANs through a comparatively low-bandwidth-wide area network (WAN). The departmental servers run Windows 2003 and Linux operating systems. Also, because of its sensitive nature, the data file is encrypted during its transfer.

As shown in the previous diagram, only the client API is installed on the DTS user computer (1). The DTS manager is installed on a separate Windows 2003 computer (2) and the DTS agent is installed on the central server (3), the departmental servers (4, 5 and 6) and all of the end systems, for a total of 300 end systems, if there are 100 end systems on each LAN.

The Windows 2003 manager machine (2) can also have CA Common Services installed; if so, it is responsible for all network, data transfer, and scheduling tasks.

The Data Transfer Problem

The problems encountered by today's business world often require solutions that involve the transfer of large volumes of critical information (data). This information can be gathered from one system and transported to many others (distribution), or it can be gathered from multiple systems and then sent to one (collection). With the added complexity of an ever-increasing number of users, often separated by thousands of miles, the ability to quickly and efficiently transfer data from one location to another becomes a difficult problem to solve.

The problem is further compounded by a corporation's enterprise, which often consists of a mixture of different computers, running different operating systems, connected through different communications links.

The task of collecting and distributing information can pose problems when you must distribute data across multiple platforms and protocols over complicated networks. Some of the more traditional problems associated with data transfer include:

Incompatibility between platforms

Transferring data stored in different formats on different platforms can often pose a problem. Difficulties arise when data cannot be transferred from one platform to another or if the integrity of the data is lost due to the incompatibilities between platforms.

Network performance

The distribution of large amounts of data across a network can often impact other network users by swamping the network with data.

Inefficient routing

The inability to determine the optimum route by which to transfer data through a large and often complex network can lead to inefficiency, resulting in lost time and effort.

Unreliability

Mission-critical data must get through. Transfers must complete successfully, and users must be aware of the success of those transfers. Data transfer products must be reliable and informative…most are not.

The Data Transfer Resolution

DTS is a comprehensive service that does the following:

- Integrates with CA solutions like Client Automation, so that you can take the CA client solution out of the box, install it, and it works!
- Provides cross-platform support. Due to its structured, yet practical architecture, porting DTS to new platforms and new operating systems is a simple task.

  Note: In Client Automation all DTS manager components (TOS, NOS, and SOS) run on Windows Servers only; managers are not supported on UNIX/Linux. For more information about supported platforms, see the Compatibility Matrix.
- Provides multi-protocol support. The DTS architecture positions the entire product above the communications protocol layer. So, the product is not dependent on any particular protocol or network technology. DTS is easily expanded to take advantage of new protocols and technologies as they become available.
- Runs over TCP, SNA, UDP, SPX, and PPP. DTS also takes advantage of both the Broadcast and Multicast network technologies to get the best out of your network.

  Note: DTS does not support the latest version of PPP communication protocol, PPPv6.
- Leverages the power of CA Common Services with which it is closely integrated. WorldView, Event Management, Workload Management, Agent Technology, and the MDB are all used by DTS to provide a world-class data transport solution.
- Provides scheduling capability. Simple or complex schedules are created and configured through the API.
- Provides transfer monitoring and control capabilities. Progress and status of all data transfers occurring in an enterprise can be monitored. Data transfers can be aborted or suspended at will. If a transfer is suspended (or if it fails), it can be resumed later.
- Provides service management facilities. Run-time parameters of all service components are altered dynamically, service components monitored, and log files viewed.
- Can be centralized or distributed; you choose! It can even be a combination of both.
- Uses client/server technology to provide a true network based interface. For example, you can use DTS to transfer data between any computers in your enterprise.
- Implements multiple-level security. User and password authentication are activated at the client, server, manager, and agent levels. Data can also be encrypted using a variety of algorithms.
- Contains an optimization engine that ensures that DTS makes the best use of your network. It works out optimal routes across your enterprise, resolves transmission settings, and decides on the best transfer mechanism to use from the specification of the data you want to transfer.

- Has a user-extensible architecture. DTS provides a fully functional SDK, and the ability to create your own agent filters. Together, these features enable you to have complete control over how you extend DTS.

Solving the Problem

In preparing for a data transfer, you should look at what you need to accomplish: What data do you need to transfer, where does it need to be transferred, and how and when should that transfer occur? Once you answer these questions, you can identify what DTS objects you must create to satisfy these needs.

This process, in overview, is quite straightforward. First, you must configure, or model, your DTS network. Second, you must create all necessary transfers, schedules, and so on.

Modeling the Network

Modeling the network involves creating logical links among the network computers to satisfy your data transport requirements. By default, when DTS is installed, it assumes that all computers are connected directly to one another. So, your first task is to model the network to satisfy your logical requirements.

Consider the sample scenario. By default, as far as Data Transport is concerned, the central server has a direct connection to all of the end systems. However, if you instruct DTS to transfer the data file from the central server to all of the end systems, by default, it would carry out a fanout transfer. The optimization engine realizes the source data file for all of the transfers is identical, so it only needs to read the data once.

Using DTS, you can carry out some simple network configuration (build a model of your Data Transport network), and tweak some object properties to let DTS take advantage of some of its advanced transfer technology. You can model our Data Transport network requirements using WorldView.

You want to stop DTS from transferring data directly from the central server to the end systems and instead transfer it through the department servers. Thus, you must hardwire a route from the central server through the department servers to the end systems. You can do this by adding Data Transport links to the network model.

First, create three links among the central server and each of the department servers. Then create three Data Transport machine container objects, one for each LAN. Place all computers that connect to the LANs (except the department servers) in their respective containers.

Next, you must create links from each of the departmental servers to their appropriate machine container object. Lastly, you want to make DTS take advantage of the point-to-many transfer mechanism when it transmits the data from the department servers out to the end systems on the local LANs; and since you are transmitting to computers connected to the same IP subnet, DTS uses the Broadcast (BCAST) protocol.

With DTS there are many ways to achieve the same result. However, it is easier to set the point-to-many protocol property of the department server to container link to BCAST and set the point-to-many network address to the broadcast address appropriate to the subnet. You can do this in WorldView, by right-clicking the link and selecting Open DTS Details from the context menu.

Choosing Open DTS Details lets you modify the following properties:

- Maximum parcel size
- Throttle factor
- Point-to-point protocol and parameters
- Fanout protocol and parameters
- Point-to-many protocol and parameters

Maximum Parcel Size

DTS transfers data in manageable chunks named parcels. The size of the parcel affects many things, including the following:

Overall transfer performance

The bigger the parcel size, the faster the transfer.

System resources the agent uses

The bigger the parcel size, the more memory an agent uses (since it has at least one parcel of data in memory at any given time during a transfer).

Checkpoint restart

After each parcel transfers, it is tagged as part of the automated checkpoint and restart feature. If a failure occurs during the transfer process, DTS can restart the transfer from the last successful parcel transfer.

System or network speed

When transferring files between agents running on slower systems or over slower networks, transfers can be suspended or can fail with the message "Connection failed to *address*|*hostname*." This condition can be due to timeouts while processing large parcels of data. If this situation occurs, we recommend reducing the maximum parcel size value for transfers to slower systems.

Throttle Factor

Throttling lets you regulate the amount of network bandwidth that is used when carrying out data transfers by specifying a delay interval between parcel sends. The throttle factor property specifies the value of the delay. Valid values range from 0 to 100. The default is 0. Each increase in throttle factor increases the delay interval by 50 milliseconds. For instance, a throttle factor value of 20 represents a one-second delay between parcel sends.

Modifying DTS Object Properties

When DTS is installed, and if CA Common Services is already installed, the Open DTS Details menu command is added to the context menu of WorldView for all Managed Objects. The Open DTS Details command lets you view read-only properties and modify DTS properties that are associated with the object that you right-clicked. The object could be a link, an interface, a machine, or a machine container.

Modifying Individual Object Properties

When you select Open DTS Details, the Object Details notebook appears, letting you modify the following properties:

- Maximum parcel size
- Throttle factor
- Point-to-point
- Fanout
- Point-to-many

Note: The Name, UUID, Class, Label, Description, and Address properties are read-only properties (information only).

By modifying these properties, administrators control how DTS performs data transfers, and thus they can configure the DTS network topology for optimum performance.

Note: Leaving a property value blank indicates that the property is undefined, and its default value is used.

Creating Transfer Objects

To encrypt a binary file during transport, read the data using a binary or text filter, apply an encryption filter, transfer the data, apply a decryption filter, and then write the data out using a binary or text filter. You could create your own filter objects in the Transfer Object Repository, but for convenience, DTS comes complete with many predefined filter objects for commonly applied data processing operations, so you can use those instead of creating our own.

Next, create transfer objects that define the data file transfer from the Solaris central server to each of the end systems. Create and configure one transfer object for each end system. These transfer objects must be the same except for the name of the receiving computer. Each transfer object must also reference the appropriate filter objects for binary read, encryption, decryption, and binary write.

You can create a transfer group object (also called a transfer job) and can add all the previously created transfer objects to it. Next, create a calendar that specifies that you can activate transfers every weekday.

Last, create a schedule object and set its time property to 22:00. Also reference the transfer group object that is previously created and the calendar, and enable the schedule object.

A DTS administrator or user can do all of the previously mentioned tasks for creating, scheduling, and activating the transfer by using the following:

- DTS dtscli command.
- DTS client API.

Note: For detailed information, see the *DTSCLI Command Reference Guide*.

More information:

Filters (see page 30)
DTS Client API (see page 33)
dtscli Command (see page 12)

Chapter 4: Configuring Data Transport Service

DTS agents and managers in Client Automation are managed by configuration policies. A configuration policy is a set of parameters that govern how a particular component behaves. Configuration policies in Client Automation can be viewed and modified from the Control Panel, Configuration, Configuration Policy, Default Computer Policy, DSM node in the DSM Explorer tree.

This section contains the following topics:

Viewing Configuration Policies

Important! It is not possible to configure the runtime parameters for Client Automation DTS agents and managers from the dialogs in the WorldView component of CA Common Services, as this would break the policy that has been set up in the DSM Explorer. The DTS WorldView Administration Client dialogs are provided for the configuration of Data Transport Service r1, r2, and r3 components only.

Configuration policy settings for the DTS components are located in the Data Transport Service (DTS) policy group under the Default Computer Policy, DSM node and appear on the Data Transport Service (DTS) pane.

These settings are propagated to the various Notebook pages and dialogs in the DTS WorldView Administration Client, which is accessible from the WorldView component of CA Common Services.

The Data Transport Service (DTS) policy group contains the following policy group folders:

- Audit
- Data Browser Agent Plugin
- Data Transport Agent Plugin
- Macro
- Network Object Server
- Schedule Object Server
- Transfer Object Server

For example, if you double-click the Data Transport Agent Plugin node, the Data Transport Agent Plugin pane displays the individual policies in that policy group, such as Agent auditing mode, Concurrency, Discreet mode, Heartbeat timeout.

More information:

Network Administration (see page 11)

Modifying Configuration Policies

You can modify the DTS configuration policies.

To change a policy setting

1. Double-click the specific policy, for example, Discreet mode.

   The Settings Properties dialog appears, enabling you to modify its value.

2. In the Value field, change the value according to your needs, and click OK.

Note: For detailed information about the DTS configuration policies and about working with configuration policies—sealing and unsealing policies, modifying policies, creating new policies, and so on—see the Configuration Policy section of the *DSM Explorer Help*.

More information:

Self Discovery (see page 13)
Network Administration (see page 11)

Chapter 5: Customizing Data Transport Service

You can customize your DTS installation by doing the following:

- Creating customized network topologies specifically for your enterprise
- Customizing messages and audit levels
- Using calendars to start and stop transfers based on time-based routing

This section contains the following topics:

Creating a Network Topology

The framework for a Data Transport Service network is created when the Install Network Business Process Views and Data Transport Auto Discovery processes are performed. It includes three Business Process Views: DTS Managers Configuration, DTS Agents Configuration, and DTS Administration. The Data Transport objects—DTS managers and DTS agents—are visually depicted in these views and become eligible for object management through the WorldView 2D Map and 3D Map.

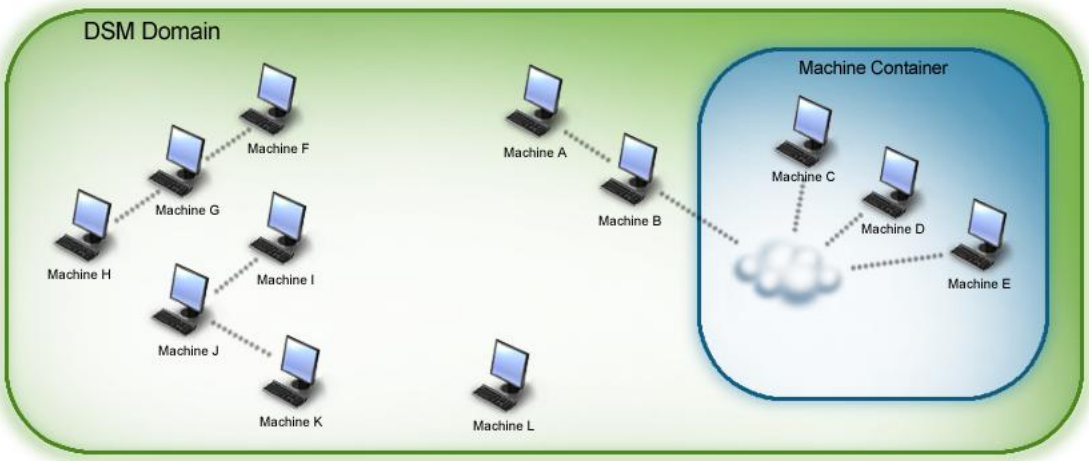
You can define the machines in your network, the network topology among the machines in this network, and the communication routes to follow when transferring data between its machines. Establishing a network topology for your Data Transport Network lets you define a strict hierarchical network structure, by which the Data Transport Service determines the routes to use for transfers between the machines of your network.

Defined machine and network properties, which are set using the DTS Administration Client interface and stored in the MDB, are responsible for establishing the hierarchical and network-related relationships between the machines of your network. These relationships enforce the route used to complete transfers through and between the machines of your network.

Machine and network properties can be set at the machine, interface, machine container, and link levels. If multiple sets of properties are encountered when attempting to resolve a route through the network, then the transfer property precedence is followed to determine the machine and network properties having the most influence over the transfer. Once determined, the transfer inherits these property values and the route is resolved.

Machines included in your network topology that are not defined in a hierarchy comprised of one or a number of links are assumed to have point-to-point connectivity to all other machines in your network.

Consider the following domain network topology:



This example defines a network topology among the machines of a Client Automation domain. Hierarchies of machines, together with machines that are not a part of any defined hierarchy, exist in the same network topology.

When considering the relationships established among the machines due to the defined network topology, you can conclude the following:

- Machines A, B and the machines of the Machine Container (C, D and E) comprise a hierarchy.
- Machines F, G and H and machines I, J and K are also examples of separate established hierarchies.
- Machines C, D and E are members of a Machine Container; you can define common transfer properties for these machines once at the container level, yet apply them to all the machines of the container. Defined Machine Container properties take precedence over the individual machine properties.
- No defined routes exist between Machines C, D and E; similarly, no defined route exists between Machines B and G. When you initiate a transfer between these machines, point-to-point, TCP/IP connectivity is assumed.
- A route from Machine A to Machine E is resolved into two routes: one from Machine A to Machine B and a second from Machine B to Machine E.

Note: For more information, see the CA WorldView documentation for Business Process Views.

Changing Audit Levels

You can enable or disable auditing for the DTS agent, TOS, NOS, or SOS by setting the appropriate configuration policy for auditing in the DSM Explorer. If you enable auditing, auditing messages are issued by the specified component when the events that trigger them occur. If you disable auditing, auditing messages are not issued by the component when the events that trigger them occur. The auditing messages (whether customized or not) are ignored but are not deleted. You can enable them again, when desired.

Note: Before you can modify a policy, you must unseal it first. For more information, see the Configuration Policy section of the *DSM Explorer Help*.

More information:

Auditing (see page 20)

Enable/Disable Agent Auditing

You can enable or disable auditing for the DTS agent by setting the appropriate configuration policy for auditing in the DSM Explorer.

To turn agent auditing on or off

1. Navigate to the Data Transport Service (DTS) subnode under the Control Panel, Configuration, Configuration Policy, Default Computer Policy, DSM node.
2. Click the Data Transport Agent Plugin policy group subnode.

   All of the policies for the DTS agent appear in the Data Transport Agent Plugin pane (right pane).
3. Double-click the Auditing mode policy on this pane.

   The Setting Properties dialog appears.
4. Set the Value field to True or False as appropriate. (The default is True.)
5. Click OK.

   The agent auditing function is enabled or disabled as specified.

Enable/Disable TOS Auditing

You can enable or disable auditing for the TOS by setting the appropriate configuration policy for auditing in the DSM Explorer.

To turn TOS auditing on or off

1. Navigate to the Data Transport Service (DTS) subnode under the Control Panel, Configuration, Configuration Policy, Default Computer Policy, DSM node.
2. Click the Transfer Object Server policy group subnode.

   All of the policies for the Transfer Object Server display on the Transfer Object Server pane.
3. Double-click Manager auditing policy on this pane.

   The Setting Properties dialog appears.
4. Set the Value field for the level of auditing detail offered by the manager, that is, the DTS Transfer Object Server. Valid values are as follows:

   0 - Disabled

   Specifies no audit messages, that is, auditing is disabled.

   1 - Minimal

   Specifies server messages only.

   2 - Standard

   Specifies server, transfer job, and transfer fail messages.

   3 - Extended

   Species all audit messages.

   Default: 1
5. Click OK.

   The TOS auditing function is enabled or disabled as specified.

Enable/Disable NOS Auditing

You can enable or disable auditing for the NOS by setting the appropriate configuration policy for auditing in the DSM Explorer.

To turn NOS auditing on or off

1. Navigate to the Data Transport Service (DTS) subnode under the Control Panel, Configuration, Configuration Policy, Default Computer Policy, DSM node.
2. Click the Network Object Server policy group subnode.

   All of the policies for the Network Object Server appear on the Network Object Server pane.
3. Double-click the Manager auditing mode policy on this pane.

   The Setting Properties dialog appears.
4. Set the Value field to True or False as appropriate. (The default is True.)
5. Click OK.

   The NOS auditing function is enabled or disabled as specified.

Enable/Disable SOS Auditing

You can enable or disable auditing for the SOS by setting the appropriate configuration policy for auditing in the DSM Explorer.

To turn SOS auditing on or off

1. Navigate to the Data Transport Service (DTS) subnode under the Control Panel, Configuration, Configuration Policy, Default Computer Policy, DSM node.
2. Click the Schedule Object Server policy group subnode.

   All of the policies for the Schedule Object Server appear on the Schedule Object Server pane.
3. Double-click the Manager auditing mode policy on this pane.

   The Setting Properties dialog appears.
4. Set the Value field to True or False as appropriate. (The default is True.)
5. Click OK.

   The SOS auditing function is enabled or disabled as specified.

Customizing Audit Messages

Data Transport Service administrators can customize messages for common events, such as the failure of a DTS agent or Transfer Object Server or a transfer job aborting. Any message that you customize replaces the corresponding default message that Data Transport Service issues under the same conditions.

For example, before customization, if the Network Object Server fails during normal operation, Data Transport Service sends the event to the Common Application Framework (CAF), which forwards a default "Network Object Server failure" message to the Network Object Server log file and the console. However, if you customize the NOS_FAILURE policy with an *audit token* in the Audit policy group in the DSM Explorer, then Data Transport Service sends your customized message instead of the default message to the console.

Thus, customized messages provide administrators with increased flexibility for handling errors. For example, if the Network Object Server resides on a machine at your site that is known to occasionally freeze and needs to be restarted, you might add that fact to the text of your customized message.

The messages you can customize are restricted to the audit tokens available in the DSM Explorer. The audit tokens apply to transfers, transfer jobs, and the servers and agents.

Note: Audit messages and macros are supplied with Data Transport Service r2 and r3. Also, you *cannot* customize other messages issued by Data Transport Service, including those regarding the protocols, filters, encryption, or operating system used in the transfer.

More information:

Auditing (see page 20)

Audit Log Files

In Client Automation, both the managers and the agents use the common event component. This means that there is no dtaaudit.log, as in previous releases. All auditing messages are directed to the Windows application log or to a specified file, but this behavior is controlled by the Client Automation event component.

Audit Tokens

Each audit token defines a message to write to the Client Automation event component, which directs it to the application log and the agent auditing log file when a specific event occurs, such as the start or completion of a transfer job. You can customize the message (called the audit value) for each token to contain free text and macros that are expanded when the token is issued, that is, when the message is issued by the Data Transport Server or Data Transport Agent.

For example, the following audit tokens represent the events for which you can customize the messages the DTS agent. The audit value is the macro plus free text, if any, that you specify for each of these events.

```
DTA_ABORT=${DT} ${TT} Transfer from ${XF} to ${XT} aborted.
DTA_COMPLETE=${DT} ${TT} Transfer from ${XF} to ${XT} completed.
DTA_FAIL=${DT} ${TT} Transfer from ${XF} to ${XT} failed; error message=${XX}.
DTA_START=${DT} ${TT} Transfer from ${XF} to ${XT} started.
DTA_SUSPEND=${DT} ${TT} Transfer from ${XF} to ${XT} suspended.
DTA_RESUME=${DT} ${TT} Transfer from ${XF} to ${XT} resumed.
AUDIT_START=${DT} ${TT} Audit file started.
AUDIT_END=${DT} ${TT} Audit file stopped.
```

The first group of audit tokens (DTA_ABORT ... DTA_RESUME) are issued when a transfer that specifies this agent as the initiator or responder aborts, completes, fails, starts, is suspended, or is resumed after a suspension.

These messages supply transfer-specific information to the initiating DTS agent. They can be very useful, because transfer-related messages from the TOS are not displayed on the agent machine unless the TOS and agent reside on the same machine.

The last two audit tokens (AUDIT_START and AUDIT_END) are issued when auditing is turned on or off.

The ${DT} and ${TT} macros used in each audit message are the short forms of ${DateStandard} and ${TimeStandard}, respectively. These macros display the date and time in the standard C library format for this DTS agent computer.

Customize Audit Tokens

You can customize audit tokens in Client Automation.

To customize the value of any of the audit tokens

1. Navigate to the Data Transport Service (DTS) subnode under the Control Panel, Configuration, Configuration Policy, Default Computer Policy, DSM node in the DSM Explorer.
2. Click the Audit policy group subnode.

   All of the DTS auditing policies appear on the Audit pane (right pane).
3. Double-click the appropriate policy, for example, DTA_ABORT.

   The Setting Properties dialog appears.
4. Edit the default text for the audit token associated with the selected policy.
5. Click OK.

   The selected audit token is customized.

The predefined audit value for the transfer-related audit tokens (DTA_ABORT ... DTA_RESUME) includes the following:

- Brief free text ("Transfer from...to...event") that describes what happens to the transfer, for example, it starts, aborts, or fails.
- The {XF} macro, the short form for {TransferFrom}, identifies the sending machine in the transfer.
- The {XT} macro, the short form for {TransferTo}, identifies the receiving machine in the transfer.

The DTA_FAIL message includes the following additional phrase:

```
...error message=${XX}.
```

The {XX} macro, the short form for {TransferExit}, specifies the reason the transfer failed. This macro displays the default message number and text for the error.

The predefined audit value for the AUDIT_START and AUDIT_END audit tokens includes the following:

- Brief free text that describes the event ("Audit file started [or stopped]...at...").
- The ${DT} and ${TT} macros, explained above.

When you customize these messages, we recommend that you retain the predefined text and macros. (The exception is the format you choose for the date and time.) By doing so, you retain important information and can easily locate the message in the DTS section of the *DSM Messages Help*, which contains possible reasons and corrective actions (if applicable) for the event. When you retain these tokens, your customization adds site-specific or company-specific value to the message, but does not remove any value.

Note: For more information, see the Configuration Policy section of the *DSM Explorer Help*.

Macros

For Data Transport Service, a macro is a set of characters that represents one or more variables or actions. You can use some macros in the audit value for any Data Transport Service server or Data Transport Agent, while other macros apply only to the servers, only to a specific server (such as the Transfer Object Server), or only to an agent. *Expansion macros* are a special class of macros that you define yourself. (The macros you can customize are restricted to those available in the DSM Explorer.)

Using macros significantly extends Data Transport Service and other CA solutions that work with it, such as Advantage Data Transport. You can specify macros for any field that can be processed as an argument or can become part of a data transfer.

Typically, you use macros for two major purposes:

- To customize the messages that are displayed on the console and written to audit log files when certain events occur involving a Data Transport Service server or Data Transport Agent.
- To help systematically alter the names of input files (when they are sent) and output files (when they are received) in data transfers.

Note: For detailed information, see the *DTSCLI Command Reference Guide*.

Customize Macros

When customizing audit values for messages, you can specify free text and one or more macros. Some macros apply to all DTS components, while other macros apply only to the servers, only to a specific server (such as the TOS), or only to the DTS agent.

To customize a macro, for example, the month and day values for the global macro modifier

1. Navigate to the Control Panel, Configuration, Configuration Policy, Default Computer Policy, DSM, Data Transport Service (DTS) node.
2. Click the Macro policy group subnode.

   All of the policies for customizing expansion macros appear on the Macro pane.
3. Double-click the appropriate macro.

   The Setting Properties dialog appears.
4. Edit the default value for the expansion macro.
5. Click OK.

   The selected macro is customized.

Note: For more information, see the Configuration Policy section of the *DSM Explorer Help*.

Chapter 6: Optimizing a Data Transfer

You must activate a transfer group for DTS to transfer data. When you activate a transfer group, the DTS optimization engine finds out the best way to achieve the data transfer. The optimization process occurs in four phases.

This section contains the following topics:

Phase 1: Network Route Analysis

Phase 1 of the transfer optimization includes the DTS network topology analysis to determine if the data should transfer directly between two machines or through one or more others (multi-hop). DTS uses the following rules to determine how to establish transfer routes through the network:

- If there is a link between source and destination machine objects.
- If the source machine object is a member of a machine container object, and there is a link defined from the source machine container object to the destination machine object.
- If the destination machine object is a member of a machine container object, and there is a link defined from the source machine object to the destination machine container object.
- If the source machine object is a member of a machine container object and the destination machine object is a member of a machine container object, and there is a link defined from the source machine container object to the destination machine container object.
- If there is no direct link between the two machines or machine container objects they are part of, then DTS examines intervening links and machine objects to determine whether there is a multi-hop route defined through the network between the two machines. If a multi-hop route is defined, it is used. If more than one multi-hop route is defined, the one with the smallest metric is used. If two or more optimal routes are found, the first one is used.
- If there is no defined route and neither the source nor destination machine objects are part of a defined topology, then the route is resolved to a direct connection from the source machine object to the destination machine object.

More information:

Network Object Server (see page 24)

Phase 2: Transfer Property Resolution

Phase 2 of the transfer optimization process includes transfer property resolution. These transfer properties determine the data transmission settings to use. Data transport properties are associated with all network objects, including link objects, machine container objects, interface objects, and machine objects. To establish the priority of the properties to use for data transfers, DTS examines these properties in the following order:

1. Link object properties
2. Machine container object properties
3. Interface object properties
4. Machine object properties
5. Default network communication properties

If there is an undefined (blank) property for an object, DTS checks the next object. So, for example, if the max_parcel_size property for a link is left blank, DTS checks the max_parcel_size property for the container object.

Some properties are GET (view-only) properties, and others are SET/GET (configurable) properties.

More information:

Network Object Server (see page 24)

Link Object Properties

If a link exists between machine objects, machine container objects, or both, then the property values of that link are used. For example, if a transfer is initiated between Machines A and D, the properties of the link between Machines A and B are consulted to determine how the first half of the multi-hop transfer is to occur, and the properties of the link between Machines B and D, for the second half of the transfer.

If there is no link between the machine objects or machine container objects included in the transfer, then the properties of the machine container object are consulted next.

Machine Container Object Properties

Machine container objects let administrators establish network properties for a group of machine objects. If, for instance, a transfer is initiated between machine objects not joined by a link and the properties of those machine objects are set through a machine container object, then the properties of that machine container object are consulted to determine how the transfer is to occur.

If neither a link nor a machine container object exist, then interface objects, and its properties, are next.

Interface Object Properties

Interface objects usually define a physical interface associated with a particular machine. Specific protocol characteristics are associated with the interface. By assigning a priority property to the interface, you can specify the order in which the interfaces are evaluated when determining the point-to-point protocol to use for a transfer. Interfaces associated with the responding machine are compared to those of the initiating machine until an interface with a matching point-to-point protocol is found.

Machines may have more than one interface associated with them. The appropriate interface is determined in order of ascending priority, depending on a match for the point-to-point protocol parameter for the initiator and responder machines.

For instance, a priority value of 1 (the highest) indicates that the interface is evaluated first. A value of 0 (the default) indicates an unassigned priority, in which case the interface is evaluated last.

When evaluating interfaces, the interfaces associated with the responding machine are compared to those associated with the initiating machine until an interface with a matching point-to-point protocol is found. So, the responding machine determines the protocol regardless of whether the responding machine is receiving or sending the data. (The responding machine sends the data when the initiating machine pulls it.)

Consider the case where a machine pulls data from another machine. In this scenario, the machine pulling (and receiving) the data is the initiator, while the machine sending the data is the responder. Regardless of data direction, the interfaces associated with the responding machine are evaluated against those of the initiating machine until a matching point-to-point protocol is found.

Communication Settings

If an appropriate interface cannot be found, then the communication settings of the individual machine objects are used to determine how the transfer is carried out. When a transfer initiates, properties of the initiating machine object determine the throttle factor to use, while the properties of the responding machine object determine the protocol. The parcel size is determined by the lower parcel size of the two computers in the transfer.

Depending on the settings of the machine object's network properties and the configuration of the links, you can use intermediate computers during the transfer. The initiating and responding computers, and subsequently the network properties referenced, change during the course of the transfer.

Phase 3: Duplicate Transfer Resolution

Phase 3 of the transfer optimization process includes the resolution of any duplicate transfers. This simple process checks the transfers in the activated transfer group to make sure that none are identical. If there are any identical transfers, they are collapsed into a single transfer. This is the case when a number of transfers are expanded into multi hops that all pass through a single, intermediate machine. The first half of the hop is usually resolved down to a single transfer. Two transfers are considered identical if the following is the same:

- Input data
- Initiating machine
- Output data
- Responding machine
- Initiator and responder security tokens (the user and password)
- All filters
- Positions in a hop sequence

  Note: A point-to-point transfer at the end of a hop sequence can never be considered a duplicate for one in the middle of a hop sequence.

More information:

Transfer Object Server (TOS) (see page 26)

Phase 4: Transfer Mechanism Selection

The final phase of the transfer optimization process, Phase 4, includes the selection of an appropriate transfer mechanism.

DTS supports three data transfer mechanisms.

- Point-to-point
- Fanout
- Point-to-many

The default is the point-to-point mechanism. DTS automatically decides which mechanism to use, by the specification of the active transfers and the configuration of the network.

When activating multiple transfers through a transfer group, use of the point-to-point mechanism involves multiple reads, multiple sends (across the network), and multiple writes for each parcel of data processed. However, fanout results in a single read, with multiple sends and multiple writes.

The most efficient mechanism is point-to-many, which means a single read, a single send, and multiple writes. However, it is not possible to use fanout or point-to-many unless the transfer objects involved have certain property values in common.

More information:

Transfer Object Server (TOS) (see page 26)

Transfer Resource Utilization

Note the following considerations when choosing transfer mechanisms:

- Broadcast is less resource-intensive than fanout.
- Fanout is less resource-intensive than point-to-point.

For a one-to-many transfer, if the parameters for broadcast are supplied (and if the transfer is not resolved to an existing transfer held in the TOS, by the duplicate transfer resolution process), then broadcast is selected preferentially to fanout, and fanout preferentially to point-to-point. This preferential ordering of protocols is independent of the precedence rules governing parameter value selection.

Relationship Between Broadcasts and Link or Container Properties

If you define broadcast parameters on a container, but not on a link to that container, it does not prevent the use of the broadcast protocol.

Point-to-Point Transfers

Point-to-point transfers are those in which data is sent from a single initiator computer to a single responder computer. This type of transfer is the simplest, because data is only being sent once to a single destination. Point-to-point transfers include the following properties:

Point-to-Point Protocol

Specifies the communications protocol to use during the point-to-point transfer.

Point-to-Point Address

Specifies the address to use for a point-to-point transfer to this machine.

Point-to-Point Parameters

Specifies any point-to-point protocol-specific parameters to take into account when performing a point-to-point data transfer to this machine.

Note: For detailed information, see the *DTSCLI Command Reference Guide*.

Fanout Transfers

Fanout transfers are those in which data is sent to multiple responding machines using a direct point-to-point communications protocol. The fanout transfer mechanism is used across any network. Point-to-point communication links are established with all of the end systems. The source data is then read once and sent many times through each link. This can increase efficiency considerably where read I/O is slow or slow/CPU bound read filtering is specified (for example, compression or encryption).

Note: You can only use the fanout transfer mechanism if the input and initiator machine for all transfer objects in the transfer group are identical.

Fanout transfers include the following properties:

Fanout Protocol

Specifies the communications protocol to use during the fanout transfer.

Fanout Address

Specifies the address to use when a fanout data transfer is sent to this machine.

Fanout Parameters

Specifies special parameters that are to be supplied to the protocol. Fanout is a transfer mechanism that transfers data by reading the data once and then sending it multiple times, once to each of the target responders.

Note: For detailed information, see the *DTSCLI Command Reference Guide*.

Fanout Transfer Criteria

You can combine two or more point-to-point transfers activated at the same time (that is, in the same transfer group) into a fanout transfer if the following are the same:

- Input data
- Output data
- Initiating machine
- Initiator security tokens (the initiator user and password) for each transfer
- Responder security tokens (the responder user and password) for each transfer
- Read parcel and read file filters
- Parcel size

Point-to-Many Transfers

Point-to-many transfers are those where data is sent to multiple responding machines using a point-to-many protocol, such as BCAST (IP Broadcast) or MCAST (IP Multicast). You can only use the point-to-many transfer mechanism on IP-based networks since the mechanism uses the features of the BCAST and MCAST protocols (which use UDP).

Important! DTS supports IPv4 broadcast and IPv4/IPv6 multicast addressing. The BCAST point-to-many protocol is only for use with IPv4 addresses. If you want to perform a broadcast-type transfer over an IPv6 network, use the MCAST protocol with the relevant IPv6 multicast address.

Note: You can only use the point-to-many transfer mechanism if the input, initiator machine, and output for all transfer objects in the transfer group are identical.

Point-to-many transfers include the following properties:

Point-to-Many Protocol

Specifies the communications protocol to use during the point-to-many transfer.

Point-to-Many Network Address

Identifies the primary network address to use when sending data transfers by way of a point-to-many protocol, like broadcast or multicast, to this machine.

Point-to-Many Parameters

Specifies special parameters to supply to the protocol. Point-to-many is a transfer mechanism that transfers data by reading the data once and sending the data once to all target responders at once.

Note: For detailed information, see the *DTSCLI Command Reference Guide*.

Point-to-Many Transfer Criteria

You can combine two or more point-to-point transfers activated at the same time (that is, in the same transfer group) into a point-to-many transfer if the following are the same:

- Input data
- Output data
- Initiating machine
- Initiator security tokens (the initiator user and password) for each transfer
- Responder security tokens (the responder user and password) for each transfer
- All filters
- Point-to-many protocol properties (are BCAST or MCAST)
- Point-to-many network address properties and are not empty

Chapter 7: Configure Network Routes for Transferring Software Package Delivery

The Software Delivery registered packages need to be transferred from one machine to another for distribution, deployment, or staging jobs. Package transfer from one machine to another uses the following technologies:

a. Data Transport Service (DTS)

b. File Transfer Pug-in.

When a SD Package is transferred between the following nodes, only DTS technology is used:

a. Enterprise Manager to Domain Manager

b. Domain Manager to Enterprise Manager

c. Domain Manager to Scalability Server

d. Domain Manager to DTS Agent with DTS as download method

The given DTS transfer combinations are on wide-area network (WAN) and customers may want to define special network routes for package transfer. CA Client Automation earlier used Worldview CCS component like 2D map for achieving the same functionality. From the current release, CA Client Automation will have in-house capability of defining the routes between two machines or set of machines when using DTS technology for transfer along with extended route properties.

Consider a use case when a network administrator wants to distribute a package from Enterprise Manager to the Domain Managers in different geographical locations. The package will generally be transferred from one continent to others. As a network administrator or CA Client Automation administrator, you would like to configure the routing nodes for the packages to be transferred from enterprise manager to each of the domain manager, which will otherwise be a point to point transfer. In addition to the routes, you may also like to configure properties such as best route, alternative routes, protocols, throttle size, maximum parcel size and so on.

For customizing the transfer routes, you must configure them first in CA Client Automation. Configure the domain manager or enterprise manager to follow specific routes and properties.

Follow these steps:

Only on Enterprise Manager and Domain Manager:

1. Navigate to Control Panel, Configuration, and the configuration policy applied on the manager machine.

2. Navigate to Data Transport Service (DTS), Transfer Object Server and set the parameter “Use CCNF Hop links” to true.
3. Restart a DTS component to pick up these new settings using the following commands:
    a. Caf stop dtstos
    b. Caf start dtstos
    c. caf stop dtsnos
    d. caf start dtsnos

Define the details of the network paths that CA Client Automation might use in transfer of the packages. DTS Manager uses this information for the network routes and properties.

Setting-up Links in CA Client Automation

You can configure the network paths by setting up the links using DSM Explorer.

Follow these steps:

1. Navigate to Control Panel, Configuration, and Configuration Views.
2. Select the DTS Hop Node configuration node and then the policy which is applied on the manager machine. Unseal the policy if required.

    The following tabs appear:

    Hop Node links

    Defines the link between the machines or a group of machines defined in the container tab. Combination of the links makes a route from a source machine to destination machine. The Administrator has the provision of defining the machine label defined in Client Automation or choosing the container defined in the container tab. Each entry in this tab also provides the link properties.

    The Administrator can create the following links/routes:

    - Machine to Machine
    - Machine to Container
    - Container to Machine
    - Container to Container

Containers

Defines a logical grouping of the machines by aliasing the computer groups defined in CA Client Automation along with the network properties associated with the group. These properties are common to all the machines defined in the group and will be considered, only if the link-level properties are not defined.

Note: Henceforth "Group Of Machines" will be referred to as Container or DT Containers in this document.

3. In the Hop node links sections, create a link between two machines or container and set the following parameters:

Source

Specifies the source machine/container of the link in question. For a single machine, machine label name is the only allowed value. Label name is a property of every computer or target in the Client Automation.

For a set of machines, configure the DT Container under the container tab, that will be populated in the drop-down list for the user selection.

Destination

Specifies the destination machine/container of the link in question. For a single machine, only label names are allowed as the values. Label name is a property of every computer or target in the Client Automation.

For a set of machines, configure the DT Container which is auto populated in the drop-down list.

Link Metric

Defines the priority of this link. Lesser the value, more is the priority. The addition of the link metric values for a particular route defines the best route, second best route and so on. For setting up the alternative routes, refer to Data Transfers, Alternative Routing section in DTS Administration Guide.

Note: If multiple links are defined from same source to destination, then the first link will be considered, irrespective of the link Metric value.

Link Throttle

Defines the delay between the parcels sent out. For an increment by 1, the delay of 50 milli-seconds is added. If set to zero, there is no delay between each parcel transfer.

Maximum Parcel Size

Specifies the maximum parcel size in bytes for this link.

Point-to-Point Protocol and Properties

Specifies the protocol that is used in point-to-point transfer. Only TCP is supported in this case. User can pass additional protocol specific parameters for the point-to-point transfer.

DTS manager decides which protocol to use based on the details of the transfer. For more details, refer to Optimized Data Transfer, Phase 4: Transfer Mechanism Selection section in DTS Administration guide.

Fan-out Protocol and Properties

Specifies the protocol that is used in Fan-out transfer. Only TCP is supported in this case. User can set additional protocol specific properties in the adjacent field.

Point-to-Many Protocol, Address, and Properties

Specifies the protocol used in case of point-to-many protocols. Currently, multi-cast and broadcast are supported.

Calendar

Defines the calendar name that is associated with this link. CCS Calendars and the new DSM Calendars of CA Client Automation are supported. The new DTS implementation always searches for the calendars in the local host calendar server.

4. Repeat the steps in point 3 for configuring all the links in your network.
5. When all the links are configured, DTS automatically picks up the best route and transfers the package accordingly. It can also try the alternative routes depending on the configuration policy defined at Configuration Policy, DSM, Software Delivery, File Transfer, DTS: Alternative routes.

Using Computer Groups in Defining Routes/Links

You can use the computer groups to define the links using DSM Explorer:

Notes:

1. CA Client Auto computer groups can be associated with the source and destination for a link.
2. To add a particular computer group as part of the hop links, create the data transport container (DT Container) of the same.

Follow these steps:

1. Navigate to Control Panel, Configuration, and Configuration Views.
2. Select the DTS Hop Node configuration node and then the policy which is applied on the manager machine. Unseal the policy, if required.
3. Go to the Container tab and set the following properties of the computer group:

Container Name

Specifies the user-defined unique container name of the data transport container.

Computer Group Name

Specifies the field associated with the computer group name defined in CA Client Automation. The name of the group should be same as it appears in GUI.

If you change the name of the group, change this parameter accordingly.

Link Throttle

Defines the delay between the parcels sent out. For an increment by 1, the delay of 50 milli-second is added. If set to zero, there is no delay between each parcel transfer.

Maximum Parcel Size

Specifies the maximum parcel size in bytes for this Container.

Point-to-Point Protocol and Properties

Specifies the protocol that is used in point-to-point transfer. Only TCP is supported in this case. User can pass additional protocol specific parameters for the point-to-point transfer.

DTS manager decides which protocol to use based on the details of the transfer. For more details, refer to Optimized Data Transfer, Phase 4: Transfer Mechanism Selection section in DTS Administration guide.

Fan-out Protocol and properties

Specifies the protocol that is used in Fan-out transfer. Only TCP is supported in this case. You can set additional properties in the adjacent field.

Point-to-Many Protocol, Address, and Properties

Specifies the protocol used in case of point-to-many protocols. Currently, multi-cast and broadcast are supported.

Calendar

Defines the calendar name that is associated with this link. CCS Calendars and the new DSM Calendars are supported. The new DTS implementation always searches for the calendars in the localhost calendar server.

Note: The properties defined at container level are considered only if the link level properties are missing.

Configure Alternative Routes

If you want to use the alternative routes, then configure them in the configuration policy of the manager.

Follow these steps:

1. Navigate to Control Panel, Configuration, and configuration policy which is applied on manager machine.
2. Navigate to Software Delivery, file transfer and change the parameter “DTS: Alternative routes”. By default, this is set to zero. Zero means only the best route will be tried. If set to 1, first alternative route after the best route will be tried, if the best route is down. If set to 2, then up to two alternative routes based on the order of the priority will be considered after the best route is tried.

Facts and Limitations

1. If the computer groups mentioned in the Container is a dynamic query-based group, it will not be evaluated automatically before transfer. User must associate engine tasks for running the queries periodically.
2. Container names/ Computer groups are not supported for intermediate or hop nodes. If set, they are ignored.
3. Link-level properties have higher precedence over the container level properties.
4. Restart TOS when the “Use CCNF Hop links” parameter changes. The behavior is undefined or Software Delivery jobs may fail.
5. If two hop links are defined between same source and destinations, then the first link is considered even if it has higher link metric value.
6. 2D-Map fails to connect to DTS repository when the “Use CCNF Hop links” is set to true. It gives the below error when we try to open the DTS details of an object.

   Failed to retrieve properties from Common Object R.
7. If the Computer label name is duplicated in Client Automation “All Computers” list and the same label is used as a hop node under the Hop link tab, then DTS Manager can pick any of the matching entry for hopping.

Chapter 8: Transfer Protocols and Mechanisms

This chapter presents an overview of the data transfer protocols and mechanisms of Data Transport Service.

This section contains the following topics:

Protocol Wrapper Interface (PWI)

DTS supports many different communications protocols and data transfer mechanisms. A thin interface layer, known as the protocol wrapper interface (PWI), isolates DTS from underlying communications protocols. PWI calls underlying protocol wrapper functions that call the appropriate protocol-specific functions. The protocol wrapper code used in the DTS ensures reliability.

Note: For detailed information, see the *DTSCLI Command Reference Guide*.

TCP

TCP protocol is inherently robust, reliable, and reasonably efficient. TCP is the best choice of communication if you have a slow, unreliable network, or perhaps noisy communications lines. DTS uses the default protocol, TCP.

UDP

The UDP protocol is fast, but unreliable; data packets can get lost and for the most part, UDP protocol does not care. UDP is a good choice if you have a good, clean, reliable, high performance network. The DTS protocol wrapper ensures that your data is reliably transferred, by re-sending lost packets of data. The use of UDP on a busy network can end up causing more problems than it solves since data packets are continually being re-sent.

IP Broadcast or BCAST

BCAST or IP broadcast is a protocol that lets an initiator send a single packet of data to multiple end systems simultaneously. The packet of data is sent to all connected computers on a particular IP subnet.

You cannot broadcast across subnets or across a WAN. DTS uses broadcast when you are sending the same source data from a single computer to the same destination location on those multiple end systems on the same local subnet. Broadcast is an efficient way of transferring data, particularly when the number of end systems is large.

Important! DTS supports IPv4 broadcast and IPv4/IPv6 multicast addressing. The BCAST point-to-many protocol is only for use with IPv4 addresses. If you want to perform a broadcast-type transfer over an IPv6 network, use the MCAST protocol with the relevant IPv6 multicast address.

Calculate a Broadcast Address

You can use the following logic algorithm to calculate a computer's broadcast address:

```
Broadcast address = (sub_net_mask AND IP_address)
OR (NOT(sub_net_mask))
```

To calculate a broadcast address

1. Translate the subnet mask and distributing system's IP address into their binary equivalents.

   For example, an IP address of 172.16.29.156 and a subnet mask of 255.255.0.0 would be expressed as follows:

   ```
   IP address = 10101100.00010000.00011101.10011100
   ```

   ```
   sub_net_mask = 11111111.11111111.00000000.00000000
   ```

   Using AND, the concatenated result is as follows:

   ```
   10101100.00010000.00000000.00000000
   ```

2. Perform a binary OR with the inverse of the subnet mask. This provides you with the broadcast address for the network to which the broadcast computer belongs, along with the appropriate subnet mask.

   Using the IP address and subnet mask from Step 1, this is expressed as follows:

   ```
   sub_net_Mask AND IP_address =
   10101100.00010000.00000000.00000000 (171.16.0.0)
       NOT (sub_net_mask)=
   00000000.00000000.11111111.11111111 (0.0.255.255)
   ```

   Using OR, together we get the following broadcast address:

   ```
   172.16.11111111.11111111 = (172.16.255.255)
   ```

IP Multicast or MCAST

IP multicast is a relatively new network technology, similar in function to broadcast, except that the end systems can be anywhere on the network. Before data is transferred, end systems join a multicast group. Data sent to the multicast address corresponding to the group is received by all members of the multicast group. A multicast address identifies a multicast group. Valid multicast addresses that you can use for DTS fall in the range 224.0.1.0 to 238.255.255.255 (Global Scope Addresses).

Note: The legacy hardware (for example, routers) may not support multicast.

DTS uses dynamic multicast groups that are removed after the transfer completes and are no longer required. Certain companies or products may already be officially registered to use multicast addresses near the bottom of the range; therefore, it is best to pick multicast addresses near the top of the range.

Important! DTS supports IPv4 broadcast and IPv4/IPv6 multicast addressing. The BCAST point-to-many protocol is only for use with IPv4 addresses. If you want to perform a broadcast-type transfer over an IPv6 network, use the MCAST protocol with the relevant IPv6 multicast address.

Reserved and Registered Multicast Addresses

The Internet Assigned Numbers Authority (IANA) maintains a list of registered IP multicast groups, which are listed in the following table:

Address	Reserved for
224.0.0.0	The base address; cannot be assigned to any group.

Address	Reserved for
224.0.0.1 to 224.0.0.255	Local Scope Addresses; used for routing protocols and other low-level topology discovery or maintenance protocols. Multicasts in this range are never forwarded off the local network, regardless of TTL. Multicast routers should not forward a multicast datagram with a destination address in this range, regardless of its TTL.
224.0.1.0 to 238.255.255.255	Global Scope Addresses; used for multicast group addresses. Some of the well-known groups include the following: All systems on this subnet (224.0.0.1) All routers on this subnet (224.0.0.2) All DVMRP routers (224.0.0.4) All OSPF routers (224.0.0.5) IETF-1-Audio (224.0.1.11) IETF-1- Video (224.0.1.12) AUDIONEWS (224.0.1.7) MUSIC- SERVICE (224.0.1.16)
239.0.0.0 to 239.255.255.255	Administratively Scoped Addresses; used for *private domains* for local site applications, *not* Internet-wide applications.

UDP Operation and Limitations

Multicast does not provide guaranteed delivery; it uses UDP to send its packets, which is less reliable than TCP. UDP is a simple, unreliable datagram protocol that supports the SOCK_DGRAM abstraction for the Internet protocol family. It is layered directly above the Protocol (IP). UDP sockets are connectionless and are typically used with the sendto, sendmsg, recvfrom, and recvmsg system calls. The following are some of the disadvantages and limitations of using multicast and UDP:

- UDP packets can be lost or discarded in several ways, including a failure of the underlying communication mechanism.
- UDP implements a checksum over the data portion of the packet. If the checksum of a received packet is in error, the packet is dropped without notifying the sender.
- UDP sockets store a limited queue of received packets. Thus, datagrams received outside of these limits are discarded without notification.

- UDP receives and processes Internet Control Message Protocol (ICMP) error messages in response to UDP packets with the following limitations:
  - ICMP “source quench” messages are ignored.
  - ICMP “destination unreachable,” “time exceeded,” and “parameter problem” messages disconnect the socket from its peer. So, later attempts to send packets using the same socket return an error.
- UDP does not guarantee that packets are delivered in the order they are sent.
- UDP may generate duplicate packets during the communication process.
- The application using UDP must verify delivery of a UDP packet. Guaranteed delivery is a key feature of DTS. The DTS agents on the initiator and responder work together to ensure proper delivery of data, even UDP packets.
- UDP packets are given low priority on a routed network. If a network is busy and needs to drop packets, the UDP packets are dropped first. When the router in a DTS environment drops a packet, the responding agent notifies the initiator that data was dropped and requests a resend. The initiator then resends the packet. This constant need for resends causes more network traffic, which causes more dropped packets.
- If a router sends multicast packets to all subnets more than once, network traffic increases drastically. Any multicast message is sent to all subnets. The routers have no way of knowing which subnets have responders to this data; therefore, the data is sent to all subnets.

Recommendations for Multicast Groups

For guaranteed delivery, we recommend keeping multicast and broadcast groups to a maximum of 50 computers per group. A reliable network can meet the demand of a multicast or broadcast transfer if the number of computers per group is below 50. However, if these groups are larger than 50, the network drops more UDP packets, therefore causing more retries. As the retry count increases, network efficiency decreases.

Network Configuration for Multicast

By default, networks disable multicast. The network administrator must ensure that multicast is enabled throughout the network. Specifically, the network administrator must enable multicast on all of the routers and switches between the initiating and responding DTS agent machines that are part of any multicast transfer.

Note: Not all routers and switches support multicast. Switches must be level two or higher to support multicast.

Platform Limitations for Multicast

Some platforms do not support multicast. Multicast support is the function of the protocol stack and the network interface on each platform. You must enable multicast, configure, or do both on some protocol stacks or network cards. For details and prerequisites, see the appropriate platform documentation.

Multicast Web Links

For more detailed information about how to configure and test a network for multicast, see the multicast information on the following Internet sources. You can also search the Internet for additional articles.

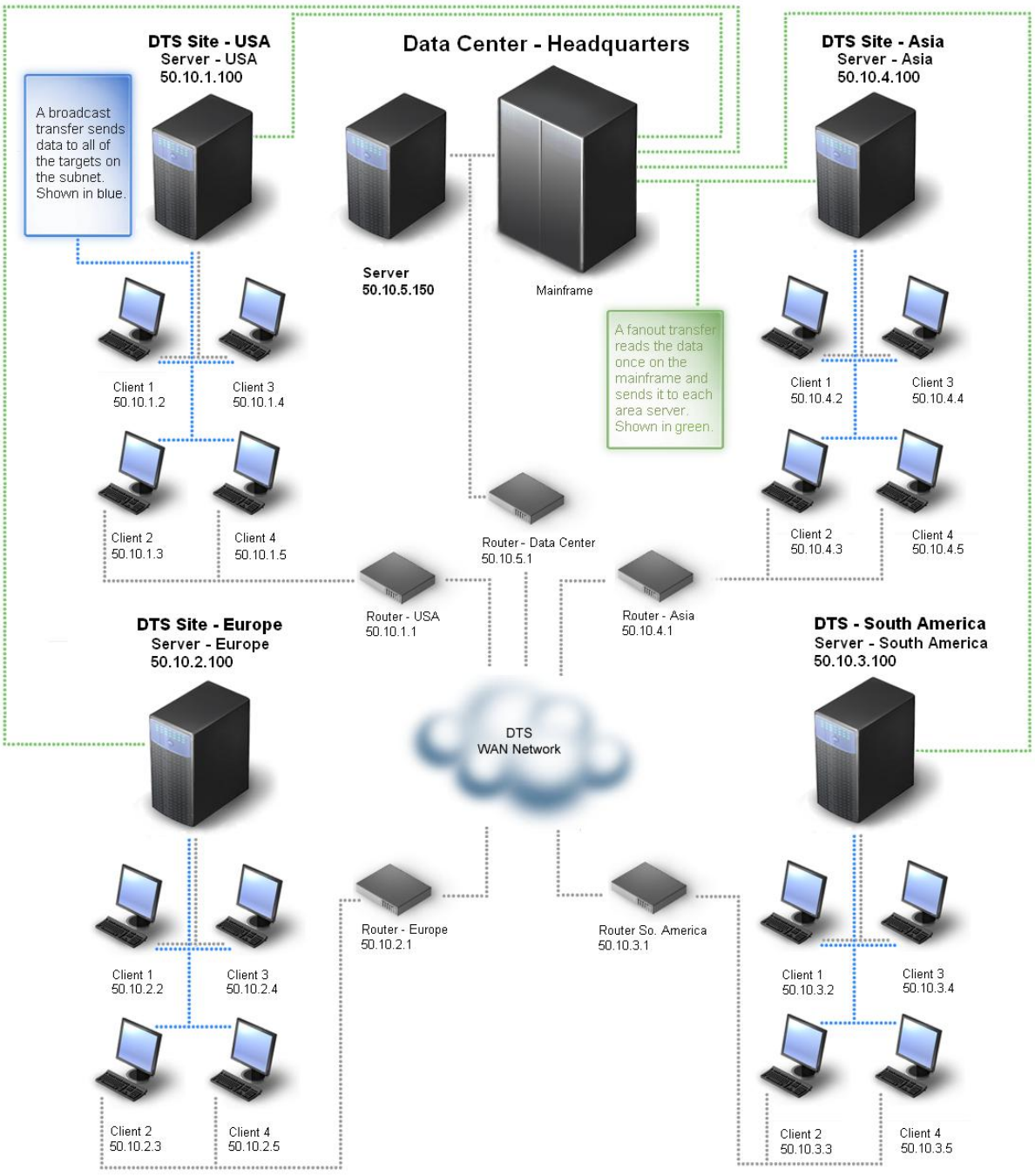
- The multicast how-to information on the Linux Documentation Project web site, http:\\www.tldp.org. This site is for Linux, but much of the multicast information applies to all platforms that use multicast.
- The Multicast Working Group web site, http:\\multicast.internet2.edu.
- The Multicast Home Page web site of the University of Southern California, http:\\netweb.usc.edu/multicast.

Using Multiple Delivery Methods

DTS supports several delivery methods and protocols; you should review all of them to determine which one or which combination is best for your environment. In most DTS networks, there are usually better alternatives to using multicast alone.

Most network configurations include a combination of point-to-point, fanout, broadcast, and multicast transfers. The following example depicts a sample DTS configuration in which the links to and from the network (cloud) represent slow 56 KB satellite connections.

DTS Multicast Example



You want to add as little traffic as possible to the network, while transferring the data to all of the desktops around the world. To accomplish this goal, you can use WorldView 2D Map to configure the computers and links in our network. Our goal is to use the most efficient delivery method for each type of transfer (global or local) in this scenario.

Note: For more information or instructions about using the 2D Map, see your WorldView documentation or online help.

Configure the Servers to Use Fanout for Cross-Continent Transfers

You can configure servers to use a fanout transfer. In our multicast example, you may want to perform a fanout from the z/OS mainframe in the data center to the server on each of the continents because the satellite link is unreliable. A fanout reads data once and writes it to each server once, which provides a much lower chance of failure than multiple simultaneous transfers.

To configure the servers for fanout

1. Open the DTS Business Process View in your WorldView 2D Map.
2. Drill-down into the Networking Administration and Monitoring folder.
3. Create a DT Container and place all the servers in it.
4. Select the container, right-click, and select Open DTS Details from the context menu.

   The Object Details dialog appears.
5. Select TCP from the drop-down list in the Fanout Protocol field. Ensure that all other fields are blank, and click OK.

   DTS is configured to use fanout each time it needs to send or receive transfers involving any machine in the container.

Using Broadcast for Local Transfers

Broadcast is the best method for server-to-desktop transfers because these transfers use a local network with no routers. When you configure DTS, specify broadcast for transfers that send data from the server to all of the desktops.

Create Containers and Links for Desktops

You can create containers and links for your DTS network.

To create containers and links for desktops

1. Open the DTS Business Process View in your WorldView 2D Map.
2. Drill-down into the Networking Administration and Monitoring folder.
3. Create multiple DT Containers called United States Desktops, Europe Desktops, Asia Desktops, and South America Desktops.
4. In each container, place the desktops that belong to that region.

5. Create separate DT Links on the 2D Map connecting each of the continent containers to the server for that continent.

   For example, create a link connecting the United States Server to the United States Desktop DT Container, and create a link connecting the Europe Server to the Europe Desktop DT Container.

6. Configure the newly created DT Links, as follows:

   a. Select each link, right-click, and select Open DTS Details from the context menu.

   b. The Object Details dialog appears.

   c. Select broadcast from the drop-down list in Point-to-Many Protocol.

   d. Enter the broadcast address in Net Addr.

   e. Enter the subnet mask of the link in Parameters.

   f. Ensure that all other fields are blank.

   g. Click OK to exit the dialog.

   Note: Repeat these steps for each link. Be sure to specify broadcast and the proper network address and network mask value.

Chapter 9: Diagnostics and Troubleshooting

This chapter includes information about diagnostic tools and several tips for troubleshooting Data Transport Service problems.

This section contains the following topics:

Log File Collection Tool dsminfo

CA Technologies provides the dsmInfo tool, which collects diagnostic information from systems that have Client Automation installed. The data collected is compressed into a single file that contains log files, system information, directory structures, and registry and environment information. This diagnostic tool is available in the Client Automation product installation media under the DiagnosticTools folder.

If a problem with Client Automation is reproducible, then run the following command to change the trace level to DETAIL:

```
cftrace -c set -l DETAIL
```

Reproduce the problem and collect the diagnostic information with the dsmInfo tool.

Notes:

For more information about this tool, see the DSMInfoReadMe.txt file available under the DiagnosticTools folder in the product installation media.

The dsmInfo tool produces ".7z" files by default. These files provide better compression than zip files, so uploading to CA Technologies is easier.

CAM Communication Not Working

Symptom:

I am experiencing a communications problem between my responder and initiator computers during DTS file transfers.

Solution:

Ensure that CA Message Queuing (CAM) has an open path through the network, by entering the following command:

```
camping hostname | IP address
```

Execute this command from the responder and initiator to verify two-way communications. If the transfer involves a TOS, run the camping command between the TOS machine and the initiator (in both directions).

CAM is used for communications between the TOS and the DTS agent, but not among any of the DTS servers (TOS, NOS, and SOS). Therefore, use the camping command only for testing communications between a TOS and its agents.

Multicast Transfer Fails (Windows)

Symptom:

How can I troubleshoot a multicast transfer error on Windows?

Solution:

Windows provides a utility, *mrinfo*, which helps with multicast troubleshooting. For instructions for using mrinfo, see your Microsoft Windows documentation and online help.

UNIX and Microsoft Utilities Note: Most diagnostic and configuration utilities are hardware-specific. The UNIX or Microsoft utility discussed here is provided with certain versions of the corresponding operating system. For further information and support regarding utilities for computers, routers, switches, and networks, see the vendor's online help and documentation.

Multicast Transfer Fails (UNIX)

Symptom:

How can I troubleshoot a multicast transfer error on UNIX computers?

Solution:

The UNIX utility, *mtrace*, helps with multicast troubleshooting. For specific information about the features of this utility and instructions for using it, see the documentation and online help for the specific UNIX platform and release you are using.

The mtrace utility prints the multicast path from a source to a receiver. mtrace uses a tracing feature of multicast routers classified as mrouted. You can access this feature through an extension to the Internet Group Management Protocol (IGMP). The mtrace utility runs a hop-by-hop trace query along the reverse path from the *receiver* to the *source*. It collects hop addresses, packet counts, and routing error conditions, and returns the results to the requestor.

For the exact syntax, see the man page for the specific UNIX platform and release you are using. Your syntax is similar to the following example.

Example

```
mtrace [ -g gateway ] [ -i if_addr ] [ -l ] [ -M ] [ -m max_hops ]
       [ -n ] [ -p ] [ -q nqueries ] [ -r resp_dest ] [ -s ] [ -S stat_int ]
       [ -t ttl ] [ -v ] [ -w waittime ] source [ receiver ] [ group ]
```

The only required parameter is the *source* host name or address.

UNIX and Microsoft Utilities Note: Most diagnostic and configuration utilities are hardware-specific. The UNIX or Microsoft utility discussed here is provided with certain versions of the corresponding operating system. For further information and support regarding utilities for computers, routers, switches, and networks, see the vendor's online help and documentation.

Business Process Views Installation Fails

Symptom:

My installation of Business Process Views failed with both repository errors and SQL errors.

Solution:

For non-English sites, the decimal symbol can cause problems when installing or upgrading Business Process Views. The following solutions apply to computers on which both Client Automation and Data Transport Service are installed. After changing the decimal symbol to a period or changing any other location-specific settings (as instructed below), run the dtsbpv program again and then check to see that the Data Transport 3D icons were installed correctly.

Windows

If your enterprise uses a comma (,) instead of a period (.) as the decimal symbol, when you run the Install Network Business Views program on your computer, Data Transport 3D icons are not added to the objects and classes in the MDB, and SQL errors occur. To fix these problems on Windows computers, select Start, Settings, Control Panel, and then select Regional Settings. In the Regional Settings dialog, set the Decimal Symbol to a period (.).

Linux (or Linux/UNIX)

If your enterprise uses a comma (,) instead of a period (.) as the decimal symbol, when you run the program to install Network Business Views on your computer, repository errors and SQL errors may occur. For instructions to change the decimal symbol to a period, or to change any other location-specific settings, see your operating system documentation.

Dial-up Support Not Working

Symptom:

How do I use DTS with a modem and dial-up provider service?

Solution:

DTS uses point-to-point protocol to establish an IP connection through a dial-up link. TCP is then used over the point-to-point link. This mechanism requires that a point-to-point provider service be available on *both* the initiator and responder computers.

Technical Support

For assistance, contact CA Support at http://ca.com/support.

Appendix A: Commands

This appendix provides summary information about DTS transfer-related CLI commands.

Note: For detailed information, see the *DTSCLI Command Reference Guide*.

This section contains the following topics:

dtacli and dtscli Commands

The following table contains brief descriptions of the dtacli and dtscli commands:

Command	Description
dtacli	Use the dtacli command to perform comprehensive point-to-many or point-to-point transfers, including the ability to perform transfers directly between agents without a Transfer Object Server or Network Object Server involved.
dtscli	The dtscli command performs all of the tasks that dtacli can, plus it includes other functions such as HTTP transfers (Internet downloads), creating and manipulating transfer jobs and schedules, and accepting input from a command file.

More information:

dtscli Command (see page 12)

dtscli Command—Manage Data Transfers

The dtscli command manages data transfers, transfer jobs, and schedules. The DTSCLI collects related arguments together in groups because of the large number of command line arguments. Arguments are specified left-to-right on the command line and are interpreted in terms of the most recently specified argument group.

This command has the following format:

```
dtscli [-tos params]
    [-sos params]
    [-transfer params]
    [-job params]
    [-schedule params]
    [-mode mode]
    [-log params]
    [-c comments-string]
    [@filename]
dtscli [-agent params]
    [-log params]
    [-c comments-string]
    [@filename]
dtscli -help [group [argument]]
dtscli -version
```

-tos

(Optional) Specifies the location of the Transfer Object Server (TOS) and any user credentials needed to connect to it.

-sos

(Optional) Specifies the location of the Schedule Object Server (SOS) and any user credentials needed to connect to it.

-transfer

(Optional) Creates, deletes, and retrieves the status of the transfer objects.

-job

(Optional) Creates, deletes, and invokes other methods on transfer job objects.

-schedule

(Optional) Creates, deletes, and invokes other methods on schedule objects.

-mode

(Optional) Specifies the object lifetime mode, controlling whether transfers created by the DTSCLI are activated after creation and, if so, whether they are deleted after the transfer completes.

-log

(Optional) Specifies whether transfers are to be activated synchronously or asynchronously, and what level of logging is required.

-c

(Optional) Specifies a comment string.

Default: None

@filename

(Optional) Specifies the full path name of the command file to be read and used as input.

-agent

(Optional) Performs agent-to-agent transfers.

-help

Displays help for the dtscli command line interface.

-version

Displays version information for the dtscli command line interface.

Example: Create Basic Transfer Operation

The following example creates a transfer object, a transfer job object, adds the transfer to the transfer job, executes the transfer job, and deletes the transfer object and the transfer job object. The ipath and rpath arguments specify the full path of the file on the initiator and responder, respectively. This example assumes the TOS is on the local machine, and both the TOS and the two DTS agents are operating in quiet mode.

```
dtscli -transfer ipath=jupiter::c:\data\03105.dat
    rpath=neptune::c:\data\jupiter\03105.dat
```

Appendix B: Support for External Filters

As mentioned earlier in this guide, DTS supports external parcel filters and external file filters.

In earlier releases of DTS, the OUTPUT= parameter for external filters lets you specify whether your external filter created no output at all or an output file. DTS extends this parameter, letting your external filter create an output directory, also.

More information:

Filters (see page 30)

Glossary

application

An *application* is a piece of software, for example, Microsoft Word.

application virtualization

Application virtualization is the encapsulation of an application, separating it from the underlying operating system on which it is executed. At runtime the application is tricked into acting as if it were directly interfacing with the original operating system and all the resources managed by it, but in reality it is not.

centrally managed environment

A *centrally managed environment* is one where the remote control domain manager controls the host settings through computer policies, global address book (GAB) items, licensing of the host agent on the domain, and user permissions. This is the default setting for CA Client Automation.

centrally managed host environment

A *centrally managed host environment* is one where either a remote control enterprise or domain manager is responsible for the configuration of the hosts and the authentication of viewer connections. It also manages the address book that users use to find hosts.

Common Configuration Enumeration (CCE)

Common Configuration Enumeration (CCE) is one of the SCAP standards. It contains Standard identifiers and dictionary for system configuration issues related to security. A rule definition in an SCAP data stream can contain references to one or more CCE identifiers, indicating that the rule is a representation of a specific CCE configuration guidance statement or configuration control. For more information, go to http://cce.mitre.org/.

Common Platform Enumeration (CPE)

Common Platform Enumeration (CPE) is one of the SCAP standards. It contains standard identifiers and dictionary for platform or product naming. For example, some elements in XCCDF files can be restricted to only apply to certain platforms and this is done using CPE identifiers. For more information, go to http://cpe.mitre.org/.

Common Vulnerabilities and Exposures (CVE)

Common Vulnerabilities and Exposures (CVE) is a dictionary of common names (that is, CVE Identifiers) for publicly known information security vulnerabilities. These identifiers make it easier to share data across separate network security databases and tools. CVE is one of the components used in SCAP. See http://cve.mitre.org/ for details.

Common Vulnerability Scoring System (CVSS)

Common Vulnerability Scoring System (CVSS) is one of the SCAP standards. It contains standards for conveying and scoring the impact of vulnerabilities. For more information, go to http://www.first.org/cvss/index.html.

configuration view

A *configuration view* is a customized Windows-only user interface that lets you edit configuration policies that are related to specific components or functionality. Configuration views summarize the relevant policies for a component or function independent of where they are actually located in the hierarchy and the DSM Explorer tree.

connectors

connectors are the links from products that consume connector data to external products, or *domain managers*. Each connector retrieves information from its domain manager and transmits the information through the connector framework to the consuming product for visualization and analysis. Connectors can also enact inbound operations on data in the source domain manager, such as object creation. connectors use a unified connector framework to enable integration with multiple consuming products.

desktop recompose

Desktop recompose is the process of assigning a new golden template to the virtual desktop. Operating systems and applications have to be maintained during their lifetime to fix problems resolved by hot fixes or service packs or to provide new features by new versions. For linked clones, this means the master image, or golden template, has to be updated. Once the updates are completed, the linked clone is recomposed and becomes active. During the recompose operation the related linked clones are linked to this new golden template and are refreshed.

desktop refresh

Desktop refresh is the process of resetting the virtual desktop to its original state. Linked clones track changes to the virtual machine with the clone. To control the storage allocations with the clone, VMware View offers the refresh operation that resets the clone to its baseline and releases all deltas provided for tracking changes. This means that all information stored to the system drive since the creation of clone or its last refresh or recompose is lost. Unlike desktop recompose, the same golden template continues to be used as before the refresh operation.

eXtensible Configuration Checklist Description Format (XCCDF)

eXtensible Configuration Checklist Description Format (XCCDF) is a specification language for writing security checklists, benchmarks, and related kinds of documents. An XCCDF document represents a structured collection of security configuration rules for some set of target computers. The specification is designed to support information interchange, document generation, organizational and situational tailoring, automated compliance testing, and compliance scoring. For more information, go to http://nvd.nist.gov/xccdf.cfm.

Federal Information Processing Standard (FIPS)

Federal Information Processing Standard (FIPS) is a security standard that is issued and approved by NIST. It specifies the security requirements that must be satisfied by a cryptographic module utilized within a security system protecting sensitive but unclassified information.

FIPS-certified cryptography module

FIPS-certified cryptography module refer to RSA CryptoC BSAFE module, which is FIPS 140-2 certified.

FIPS-Compliant Cryptography

FIPS-compliant cryptography refers to the use of FIPS 140-2 certified modules, FIPS-approved, and FIPS-allowed techniques and algorithms for cryptography.

FIPS-only

FIPS-only is a mode of operation for Client Automation wherein only FIPS-compliant cryptography is allowed. In this mode, Client Automation is not backward compatible with the previous releases of Client Automation.

FIPS-preferred

FIPS-preferred is a mode of operation for Client Automation wherein bulk of cryptographic operations are FIPS-compliant, leaving few encryptions in legacy format. In this mode, Client Automation is backward-compatible with the previous releases of Client Automation.

golden template

In Client Automation terminology, the *golden template* is the virtual machine from which virtual desktops are cloned.

guest

A *guest* in generic platform virtualization terminology is the virtual machine and the guest operating system.

guest operating system

The *guest operating system* is the operating system running inside a virtual machine.

health monitoring

Health Monitoring (HM) functionality lets you configure alerts, set threshold values, and monitor the overall health of the Client Automation infrastructure.

host

A *host* in generic platform virtualization terminology is the physical machine, the host operating system, and the hypervisor.

host cluster

The *host cluster* is the aggregate computing and memory resources of a group of hosts sharing some or all of the same network and storage.

host operating system

The *host operating system* is the operating system running on a physical machine.

hosted virtual environment

A *hosted virtual environment* is the virtualization software that runs on top of a host operating system, that is, the physical machine, host OS, and the hypervisor.

hypervisor

The *hypervisor* is the virtualization software layer simulating physical hardware on behalf of the guest operating system. This term is synonymous with Virtual Machine Monitor (VMM).

instance software state database

The *instance software state database* is a part of the software state database that contains the history of all software jobs executed by the agent running on a non-golden template system, that is, any clones of the golden template.

linked clones

In VMware View, *linked clones* of a master or golden image only refer to the master or golden image but do not include it. Changes to the system during user sessions are not stored to the master image but are kept in delta files with the clone.

location awareness

Location Awareness lets DSM Agent on a computer detect the location of the computer.

master target device

In Citrix XenDesktop, a *master target device* is the base desktop with the OS and required set of applications from which a vDisk is generated.

master vDisk

In Citrix XenDesktop, a *master vDisk* is the initial vDisk generated from the golden template machine.

MITRE

The *MITRE Corporation* is a not-for-profit organization chartered to work in the public interest. MITRE offers the interpreters, source code, schemas, and data files at no cost so that individuals and organizations can build and expand upon them. Ovaldi is one such interpreter that is freely available.

National Institute of Standards and Technology (NIST)

National Institute of Standards and Technology (NIST) is a non-regulatory federal agency within the U.S. Department of Commerce. NIST's mission is to promote U.S. innovation and industrial competitiveness by advancing measurement science, standards, and technology in ways that enhance economic security and improve our quality of life. The United States (U.S.) National Vulnerability Database (NVD), operated by the NIST, provides a repository and data feeds of content that utilize the SCAP standards. It is also the repository for certain official SCAP standards data. Thus, NIST defines open standards within the SCAP context and defines the mappings between the SCAP enumeration standards.

native virtual environment

A *native virtual environment* is the virtualization software that runs directly on the physical machine, becoming or acting as a host operating system (often minimal), that is, the physical machine and the hypervisor. A synonymous term is "bare metal environment."

non-linked clones

In VMware View, *non-linked clones,* or full clones, are full copies of a master or golden image. The clone includes a copy of the image and all changes to the system during user sessions are stored to this copy.

nonpersistent clones

Nonpersistent clones are virtual desktops from the nonpersistent pool of VMware View user data that are transient out-of-the-box. Once a user logs off, the clone is refreshed and all user data at the system disk are lost.

nonpersistent linked clone virtual desktop

A *nonpersistent linked clone virtual desktop* is a virtual machine that is refreshed or recomposed every time the user logs on, with no persistence for custom installed applications, personalization, and so on.

offline patching

Offline Patching lets you export the patch content and patch files remotely and import to the Client Automation environment using CA Patch Manager without accessing Internet.

Offline RAC

Offline RAC is a reinstall after crash (RAC) task that is driven by the agent rather than by the manager. Virtual desktops are *recomposed* frequently, that is, whenever the golden template is updated and the disk is reset, any changes to the virtual desktop since the previous reset are effectively voided. For virtual desktops, the agent and not the manager is responsible for the creation of the RAC job container. When the disk reset occurs, the agent initiates an Offline RAC to restore any software that has been deployed to the agent.

Open Vulnerability and Assessment Language (OVAL)

Open Vulnerability and Assessment Language (OVAL) is one of the SCAP standards. It contains standard XML for testing procedures for security related software flaws, configuration issues, and patches as well as for reporting the results of the tests. All the rule checks in the checklists take the form of references to OVAL definitions contained in OVAL files from the SCAP data stream. For more information, go to http://oval.mitre.org/.

Ovaldi

Ovaldi is an OVAL Interpreter developed by the MITRE Corporation. It is a freely available reference implementation created to show how information can be collected from a computer for testing to evaluate and carry out the OVAL definitions for that platform, and to report the results of the tests. The interpreter demonstrates the usability of OVAL Definitions and ensures correct syntax and adherence to the OVAL Schemas.

package format

The *package format* is a property of a software package. Formats include regular and virtual.

package type

The *package type* is a property of a software package. Current types include Generic, MSI, SXP, PIF, and PKG. Package type is not used or altered for the purpose of supporting virtual application packages.

partition

A *partition* is an isolated instance of a host operating system. Partitions do not usually use guest operating systems because they all share the host's operating system.

partitioned virtual environment

A *partitioned virtual environment* is one where multiple instances of the host operating system can run in isolation on the same physical machine. This is not strictly a virtualization technology, but is used to solve the same type of problems.

persistent clones

Persistent clones are virtual desktops from the persistent pool that survive as they are after the user has logged off until they are refreshed or recomposed. VMware View offers out-of the box separate devices for system and user data with the persistent clones. Information stored to the user data device survives any refresh or recompose action while changes to the system disk are lost.

persistent linked clone virtual desktop

A *persistent linked clone virtual desktop* is a virtual machine that is dedicated to a specific user, and the user can request specific software to be added, customize settings, and so on. At each logon the user's customized environment is restored. This persists until the virtual desktop is refreshed or recomposed. At that point, all the software products installed on system drive are lost.

persistent non-linked clone virtual desktop

A *persistent non-linked clone virtual desktop* is a virtual machine that is dedicated to a specific user and is presented to that user at each logon with their custom installed applications, user settings, data, and so on.

platform virtualization

Platform virtualization is the encapsulation of computers or operating systems, hiding their physical characteristics from users and emulating the computing platform at runtime.

provisioned application

A *provisioned application* is an application (regular or virtual) that has been made available for execution on a target computer. The application need not be "installed" locally in order to treat it as provisioned.

regular application

A *regular application* is application software that has not been virtualized and can be installed and executed in a traditional fashion. When talking about releases, patches, and suites, regular applications are implied.

Replication

Replication is an engine task to perform the data replication from Domain Manager to Enterprise Manager and Enterprise Manager to Domain Manager.

sandbox

A sandbox is an application runtime environment that isolates the application from the computer's operating system and resources and also from other applications on the computer. The degree of isolation is usually set to allow the application some access to the operating system resources, such as the documents folder.

scalability server

A *scalability server* is the central server to enable geographical scalability for management tasks. It is a distributed process that is the primary interface for agents.

SCAP data stream

SCAP data stream consists of security checklist data represented in automated XML formats, vulnerability and product name related enumerations, and mappings between the enumerations. An SCAP data stream consists of the XML following files:

- An XCCDF file
- One or more OVAL files
- (Optional) A CPE dictionary file

schema map

A *schema map* is a mapping of the attribute names associated with data objects, such as users, computers, and groups, used in an external directory to those attribute names used by corresponding Client Automation objects. The fixed and standard set of DSM attribute names is used for querying directories and for formulating complex queries and reports.

Security Content Automation Protocol (SCAP)

The *Security Content Automation Protocol (SCAP)*, pronounced "S Cap", is a method for using the standards such as XCCDF, CCE, CVE, CVSS, CPE, and OVAL to enable automated vulnerability management, measurement, and policy compliance evaluation (e.g., FISMA compliance). More specifically, SCAP is a suite of selected open standards that enumerate software flaws, security related configuration issues, and product names; measure systems to determine the presence of vulnerabilities; and provide mechanisms to rank (score) the results of these measurements in order to evaluate the impact of the discovered security issues. SCAP defines how these standards are combined. The National Vulnerability Database provides a repository and data feeds of content that use the SCAP standards. For more information, go to http://nvd.nist.gov/.

software signature

A *software signature* defines the attributes of a software application, such as the main executable file name, other associated files, size range, version range, creation, and modification dates of the software. All these attributes of a software signature uniquely identify a software application. Software signatures in asset management are created as software definitions. You can create software definitions for a product, release, patch, suite, suite component, or virtual application image. By default, asset management provides predefined software signatures covering the most widely used software in the IT industry.

software type

The *software type* is a property of a software definition. Current types include suite, product, release, patch, and virtual application image.

staged virtual application image

A *staged virtual application image* is a virtual application image that has been discovered in the file system of a computer.

stand-alone environment

A *stand-alone environment* is one where the users of the host and viewer computers locally manage all settings, properties, and licensing of the Client Automation remote control component. It is set by a Standalone Agent installation. To install it manually, the RC agent setup needs to be called directly.

standalone virtual application

A *standalone virtual application* is a virtual application that has been provisioned in a way where the virtual application image used as the source resides on the system to which it has been provisioned.

streamed virtual application

A *streamed virtual application* is a virtual application that has been provisioned in a way where the virtual application image used as the source resides on a remote system that is different from the system to which it has been provisioned.

streamed virtual application image

A *streamed virtual application image* is a virtual application image that has been discovered to be accessible through the network from a computer. Discovery of streamed virtual application images will usually only be possible if the virtual applications residing inside of the image have been provisioned.

vDisk

In Citrix XenDesktop, a *vDisk*, or virtual disk, is basically an image file with the OS and the required set of applications.

virtual application (VA)

A *virtual applicatio*n is software that has been virtualized.

virtual application image

A *virtual application image* contains one or more virtual applications stored inside a file, possibly with a set of supporting metadata files.

virtual application image definition

A *virtual application image definition* describes the "footprint" for discovering a virtual application image. To discover an image containing one or more included virtual applications (stored inside), regular software signatures must be associated with the virtual application image definition.

virtual application package (VAP)

A virtual application image packaged inside of one or more software delivery packages is referred to as a v*irtual application package*. These packages are used to provision computers with virtual applications.

virtual application staging package

A *virtual application staging package* is a virtual application package used to stage the virtual application image.

virtual application standalone package

A *virtual application standalone package* is a virtual application package used to provision a virtual application in standalone mode.

virtual application streaming package

A *virtual application streaming package* is a virtual application package used to provision a virtual application in streaming mode.

virtual disk

A *virtual disk* is a set of files that forms a file system that appears as a physical disk to the guest operating system.

virtual image

A *virtual image* is a file or set of of files containing the complete definition of a virtual machine, including its hardware specifications and virtual disks. It is the host's file system representation of a guest. A virtual image can be online or offline depending on the running state of the virtual machine it captures.

virtual machine (VM)

A *virtual machine* is an isolated virtualized environment simulating a physical machine. The virtual machine does by definition not include the guest operating system.

virtual patch

A *virtual patch* is the virtual equivalent of a regular patch and has the same basic meaning. The term is used when reporting software inventory for virtual applications (not virtual application images).

virtual release

A *virtual release* is the virtual equivalent of the regular release and has the same basic meaning. The term is used when reporting software inventory for virtual applications (not virtual application images). Note that a provisioned virtual application can use either a staged or streamed virtual application image as source. The virtual applications contained within the virtual application image can themselves be seen as staged but not yet provisioned.

XCCDF profile

An *XCCDF profile* is a policy that is applied to the target computer or compared to the configuration of the target computer. The XCCDF file for each SCAP data stream defines the list of profiles supported. The XCCDF file must have at least one XCCDF profile, which specifies the rules to be used for checking a particular type of system. You can create separate XCCDF profiles for each applicable operational environment in which a system may be deployed.

Index