

Tru64 UNIX Best Practice

Launching a COM Application

May 2001

Product Version: COM for Tru64 UNIX Version 1.1 or higher

Operating System and Version: Tru64 UNIX Version 4.0D and higher

This Best Practice describes how to launch a Component Object Model (COM) server application on Tru64 UNIX.

Contents

Launching a COM Application on a Tru64 UNIX Server

Is This Best Practice Right for You?	1
Before You Begin	2
Establishing a COM Run-Time Environment	2
Establishing Authentication Security	3
Applying the Best Practice	4
Verifying Success	7
Troubleshooting	7
Alternative Practices	10
Comments and Questions	10
Legal Notice	11

Launching a COM Application on a Tru64 UNIX Server

This Best Practice describes how to launch a Component Object Model (COM) server application on the Tru64 UNIX operating system. It summarizes the steps for:

- Installing and initializing the COM Run Time
- Compiling and linking a COM Interface Definition Language (IDL) file into an executable COM application
- Registering and launching the COM server application from a remote NT client

See the Tru64 UNIX Best Practices Web page for more information about Best Practices documentation.

Is This Best Practice Right for You?

This Best Practice does not describe COM, nor does it fully describe the implementation of COM on Tru64 UNIX. If you are unfamiliar with COM development, see the Microsoft MSDN library or third-party documentation. If you want details on the COM Run Time and its implementation on Tru64 UNIX, see the product documentation set on the COM for Tru64 UNIX Web site.

Not all Best Practices apply to all configurations, so you must be sure that it is appropriate for your system and circumstances. To use this Best Practice, you must meet the requirements described in the following table:

Requirement	Description
COM Software version	COM for Tru64 UNIX Version 1.1 or higher, available on the Tru64 UNIX Version 5.0A or higher Associated Products CD-ROM.
Server Operating System	Tru64 UNIX Version 4.0D or higher.
Server System Configuration	128 MB minimum RAM and Tru64 UNIX C++ compiler, Version 6.2 or higher.

Requirement	Description
Client Operating System	One or more Microsoft Windows NT systems with Version 4.0 and service pack 3 or higher.
Client System Configuration	Clients with a network connection to the Tru64 UNIX server, TCP/IP enabled, and the server node name and IP address registered.

If you do not meet the previous requirements, see *Alternative Practices* for information.

Before You Begin

Before you apply the Launching a COM Application Best Practice, you must:

- Install the COM Run Time and Development Environment on your Tru64 UNIX system
- Initialize the run-time daemons and COM Registry
- Optionally, you can enable authentication security

Ordinarily, the system administrator installs and activates the COM Run Time and Development Environment and establishes security between the server and clients. These tasks are usually part of the COM for Tru64 UNIX product installation described in the COM for Tru64 UNIX Installation Guide.

Establishing a COM Run-Time Environment

Every Tru64 UNIX machine (client or server) that runs a COM application must have the COM Run Time active. The Run-Time Environment includes:

- A running NT daemon (`ntd`).
- A running Private Authentication Layer daemon (`paulad`) and a running Remote Procedure Call daemon (`rpcss`).
- A reserved network communication link and a running helper application (`coolrip`) that supports `rpcss` access to the communications port. The `rpcss` service automatically starts `coolrip`.

- An initialized Registry that contains registration information on the applications you intend to run.
- A basic COM configuration file (`/etc/dcomconfig`) that defines the environment variables and configures the environment for running remote COM applications.

COM for Tru64 UNIX includes a COM setup script, `dcomsetup`, that you use to establish the Run-Time Environment. The `dcomsetup` command displays a menu. From the menu, select *Initialize* to initialize the COM Registry, start and stop `ntd`, `paulad`, and `rpcss`, and establish a reserved network communication link and a basic COM configuration file.

Establishing Authentication Security

Security is optional and requires the `paulas` service running on the NT Primary Domain Controller, ensures authenticated user communications between `paulad` running on the Tru64 UNIX server and clients. Security is available with COM for Tru64 UNIX Version 1.1 and higher.

You can use COM with security disabled, as described in *Alternative Practices*.

To activate authentication security, the system administrator must do the following (these steps are more fully explained in the COM for Tru64 UNIX Installation Guide):

1. Use `ftp` in binary mode to copy the `paulas` executable file from its installed location on Tru64 UNIX (`nt/paulas.exe`) to the Windows NT Primary Domain Controller (the machine where authenticated communication between `paulas` and `paulad` is used).
2. Include the Domain Controller's IP address or host name in the `COOL_PAULA_DC_ADDRESS` environment variable in the `/etc/dcomconfig` file.
3. Include the Domain Controller's domain name in the `COOL_PRIMARY_DOMAIN` environment variable in the `/etc/dcomconfig` file.
4. Install a `paula.txt` file, containing the same password, on both the server (`/etc/com/paula.txt`) and Domain Controller (`%SystemRoot%\System32\paula.txt`). The `paula.txt` file on the server must be owned by the user who started `ntd`.

5. On the Windows NT Primary Domain Controller, register the user name and password of each client or server COM user who will call a remote object on the server.
6. Start `paulas`. Under the Windows NT Primary Domain Controller Control Panel, click on *Services*. If `paulas` is correctly installed, the Services display includes *Cool Paula Service*. Select *Cool Paula Service*.
7. Click the *startup* button. Be sure that the startup type is *Automatic* and that *Log On As System Account* is checked.

The `dcomsetup` menu *Verify* option uses a sample application to test the Run Time services and security. To verify a security-enabled environment, `paulas` must be started and client user names and passwords must be registered on the Primary Domain Controller, Run-Time services must be started on the server, and the `/etc/dcomconfig` file must contain the name and IP address of the Primary Domain Controller.

Applying the Best Practice

Before you compile, register, and launch the COM application, be sure to follow the recommendations in *Before You Begin*.

In general, use the following steps to compile, link, and register a COM application for remote execution on Tru64 UNIX:

1. Define the COM object's interfaces with the Interface Definition Language and create an IDL file (`.idl`). Also create a registration file (`.reg`) containing Globally Unique Identifiers (GUIDs) for the interfaces.
2. Use the Microsoft Interface Definition Language (MIDL) compiler to process the IDL file. The MIDL compiler generates C++ compatible modules (client and server source files, header files, and, optionally, an application configuration file) that you can compile into a COM executable, shared library, or type library.
3. For a proxy and stub library (`.so`), create a Module Definition File (`.def`) file that specifies the library name that exports the `DllRegisterServer`, `DllUnregisterServer`, `DllCanUnloadNow`, and `DllGetClassObject` functions. Use the `makedef` utility to compile the `.def` file into a file suitable for input to the C++ compiler.
4. Use the C++ compiler to compile and link the source and header files and required libraries into proxy and stub libraries or executables. By default, the compiler produces an executable, which is required

for remote application servers. You can choose to have the compiler produce a shared library (equivalent to a Microsoft Dynamic Link Library) that can be used as an in-process server or, when associated with a DLL surrogate process, as a remote server.

5. Copy an existing registration file (using the `sermon reg import` command), or create a new registration file, containing the GUIDs for the proxies and stubs or type libraries. Use the `sermon reg commit` command to merge the file into the server COM Registry. To register a shared library, use the `regsvr` utility.

To compile and link a COM application on Tru64 UNIX, you must include the following MIDL compiler options, C++ compiler and linker options, shared libraries, and compiler flags:

- MIDL Compiler options:

- Zp8 (packing level)
- char unsigned (data type)
- ms_ext (Microsoft extensions)
- c_ext (Microsoft C extensions)
- Os (optimization)

- C++ Compiler and Linker options:

- g (debugging)
- inline manual (inline expansion of function calls)
- ms (allows Microsoft C++ ANSI constructs)
- unsigned (consistent char declarations)
- shared (shared library) or the default -call_shared (executable)
- error_unresolved (unresolved symbol failure)
- depth_ring_search (symbol resolution method)
- pthread (use thread-safe library)

Use `-pthread` only if your COM application makes `pthread` calls. If you do use `-pthread`, do not specify the system default `-lc` and `-lcxx` libraries.

- Shared libraries:

- libmutant.so (Win32 system services)
- libmutantstubs.a (Win32 client support)
- libole32.so (basic API support)
- librpcrt4.so (distributed transport layer)
- libntrtl.so (NT executive)

```
-libcoolmisc.so (miscellaneous NT functions)
-liboleaut32.so (optional, except for generating type library
or for automation APIs)
```

- **Compiler flags:**

```
-DSAG_COM=1 (platform-specific code)
-DACD_DIGITAL_UNIX (Tru64 UNIX public code)
-DCE_TAXPOSF1 (platform-specific code)
-D_WIN32_WINNT=0x400 (Windows NT Version 4.0)
-DINC_OLE2 (OLE automation)
-DUNICODE (wchar_t Unicode conformance)
-DREGISTER_PROXY_DLL (required for DLLs)
-DCOBJMACROS (required for DLLs)
-DWIN32=100 (MIDL requirement)
```

To make your server application (the compiled and linked library or executable) known to clients, you must register the application on the server that hosts your application and on the system(s) that host the calling client applications. Once the information is in the Registry databases of the client and server machines, the client application is able to call the server and launch or access the server application.

A registration file (.reg) is an ASCII text file that contains your application's configuration information.

Normally, when you create a registration file on Tru64 UNIX, you copy lines from an existing registration file into a new file and modify that new file to meet your application's requirements. You then use the `sermon reg import` command to merge the registration file into the COM Registry. The `sermon` utility uses the key and subkey information you provide in the registration file to determine where to place the data in the Registry.

The default security level established when a client application issues a call to launch or access a server application depends on the presence or absence of a `CoInitializeSecurity` call in both the client (calling) application and the server (called) application. The security level is established on the following basis:

1. If the client application does not issue a `CoInitializeSecurity` call or the call contains no explicit setting, the security level defined in the Registry determines the default security level for that process.
2. If the client application does not issue a `CoInitializeSecurity` call, or the call does not have an explicit setting and there is no

default security level defined in the Registry, CONNECT is the default authentication level.

3. If the server application contains a `CoInitializeSecurity` call, those settings become the minimal setting for security. If the client application's `CoInitializeSecurity` call setting differs, the higher security level applies.
4. Likewise, if the server application does not issue a `CoInitializeSecurity` call, the Registry settings become the minimal setting for security. If the client application's `CoInitializeSecurity` call differs from the Registry security settings, the higher security level applies.

To start the server or client application on Tru64 UNIX, you invoke the application name from the command-line prompt and, optionally, append the options defined for your application in its code.

Verifying Success

To verify the installation and configuration of the COM Run-Time Environment, use the `dcomsetup` menu *Verify* option. If correctly installed and configured, `dcomsetup` returns "success". Remember that verification of a security-enabled server requires a Primary Domain Controller that has `paulas` running, is defined in `/etc/dcomconfig` and contains registered client user names and passwords.

The COM for Tru64 UNIX documentation describes the various error messages that indicate a compilation failure or the failure to successfully launch or access a server application.

If this Best Practice is not successful, see *Troubleshooting* for information about identifying and solving problems.

Troubleshooting

If you determine that the Best Practice was not successful, as described in *Verifying Success*, use the following table to identify and solve problems:

Problem	Possible Solutions
The <code>ntd</code> daemon won't restart.	If you use the Tru64 UNIX <code>kill</code> command to stop the <code>ntd</code> daemon process, the Registry is not updated and the <code>ntd</code> files can be left in the server <code>/tmp</code> directory. If this is the case, an attempt to restart <code>ntd</code> will fail. Remove the files starting with <code>COOL_NTD</code> from the server <code>/tmp</code> directory.
COM applications will not compile on the Tru64 UNIX server.	<p>Is the operating system Tru64 UNIX Versions 4.0D higher?</p> <p>Is the C++ compiler Version 6.2 or higher?</p> <p>Are all the required C++ compiler and linker options specified?</p> <p>Are all the required shared libraries linked to the application? Is the path (<code>/usr/shlib</code>) correct?</p> <p>Are all the required compiler flags specified?</p>

Problem	Possible Solutions
Applications will not register on the server.	<p>Is the COM for Tru64 UNIX Run Time (<code>ntd</code>, <code>paulad</code>, and <code>rpcss</code>) installed and active?</p> <p>Remember that an application can self-register (if it is coded to do so) or you can manually register an application by importing the application's <code>.reg</code> file and committing the <code>.reg</code> file to the server Registry. You can also use <code>sermon</code> to add application information to the Registry.</p>
Client applications that attempt to launch the server application return "Access Denied" errors.	<p>Do the owner and group permissions of <code>ntd</code> on the server match those required by <code>rpcss</code> when it issues a launch or access call on the client's behalf?</p> <p>Did you start <code>ntd</code> under root? Because <code>ntd</code> takes on the permissions of the UID under which it starts, you must not start <code>ntd</code> under root.</p> <p>Are the security settings for the client application and the server application compatible? To bypass security checks, set both the client and server application <code>CoInitializeSecurity</code> authorization level to <code>NONE</code> and impersonation level to <code>ANONYMOUS</code>.</p> <p>Is <code>/etc/com/paula.txt</code> on the server owned by the same user that installed <code>ntd</code> and are read and write permissions restricted to that user ID?</p> <p>Are <code>paulad</code> and <code>paulas</code> running?</p> <p>Is the <code>COOL_PAULA_DC_ADDRESS</code> environment variable in <code>/etc/dcomconfig</code> updated for the IP address or host name of the Windows NT Primary Domain Controller?</p> <p>Does the password file, <code>paula.txt</code>, exist on both the server and the Windows NT Primary Domain Controller? Does the file contain the same password?</p> <p>Are client application users and the server application user registered on the Windows NT Primary Domain Controller? Each client and server user name and password must be the same as the user name and password on the Tru64 UNIX server.</p>
	<p>Did you reboot the Windows NT Primary Domain Controller after <code>paulas.exe</code> was installed?</p>

Alternative Practices

Although this Best Practice is the recommended method for launching a COM application on a Tru64 UNIX server, there are other methods for configuring the COM Run Time, compiling COM applications, launching applications without security and launching without a Primary Domain Controller.

You can establish a Windows NT client and Tru64 UNIX server environment that allows you to use an NT server or workstation as an alternative to a Primary Domain Controller, and issue remote network calls that bypass the paulad and paulas security check. The steps you use to establish such an environment are as follows:

1. Enter the NT machine's IP address in the `/etc/dcomconfig` file `COOL_PAULA_DC_ADDRESS` variable.
2. Enter the word `workstation` in the `/etc/dcomconfig` file `COOL_PRIMARY_DOMAIN` variable.
3. Make sure that both the server application and the calling client application contain `CoInitializeSecurity` calls with `RPC_C_AUTHN_LEVEL` set to `NONE` and `RPC_C_IMP_LEVEL` set to `ANONYMOUS`. If either the client or server application is set to a higher level of security, the call fails.
4. Run `dcomconfig` on the NT client and enable *Distributed COM*, set the *Default Authentication Level* to `NONE`, and set the *Default Impersonation Level* to `ANONYMOUS`.

Under this type of environment, users registered on the remote NT machine specified in `COOL_PAULA_DC_ADDRESS` can issue calls that launch or access a running server COM object. The calls will bypass the paulad and paulas security mechanism and avoid any user name and password authentication.

These alternatives and details on this Best Practice are described in the COM for Tru64 UNIX Building and Running Server Applications manual.

Your system must meet the requirements described in *Is This Best Practice Right for You?*.

Comments and Questions

We value your comments and questions on the information in this document. Please mail your comments to us at this address:

`best_practices@zk3.dec.com`

Legal Notice

Compaq and the Compaq logo Registered in U.S. Patent and Trademark Office. Tru64 is a trademark of Compaq Information Technologies Group, L.P. in the United States and other countries. UNIX is a trademark of The Open Group in the United States and other countries. Microsoft, Windows, Windows NT are trademarks of Microsoft Corporation in the United States and other countries. All other product names mentioned herein may be trademarks of their respective companies.

Confidential computer software. Valid license from Compaq required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Compaq shall not be liable for technical or editorial errors or omissions contained herein. The information in this document is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for Compaq products are set forth in the express limited warranty statements accompanying such products. Nothing herein should be construed as constituting an additional warranty.