Unity IP Fax Outbound Troubleshooting

Document ID: 63355

Contents

Introduction

Outbound fax from a Cisco Unity user to a fax machine on the public switched telephone network (PSTN) is supported with the Cisco Unity IP Fax Configuration Wizard that starts with Cisco Unity 4.0(4) and Cisco IOS® Software Release 12.3(7)T. This service allows users to send outbound faxes through a simple e−mail. The destination fax phone number is included in the Subject line of the e−mail and is sent to a mailbox that the Cisco Unity IP Fax service checks. The service reformats the message, renders all attachments in the proper format, re−addresses, and sends the message. The message is sent through Simple Mail Transfer Protocol (SMTP) to the Cisco IOS gateway configured with the T.37 OffRamp feature in order to send the fax to the destination phone number.

This document discusses the common steps you use to troubleshoot. Refer to the Cisco Unity IP Fax Outbound Gateway Configuration Example for the initial configuration steps of the Cisco IOS gateway. Refer to the IP Fax Configuration Wizard for Cisco Unity.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Regular documentation for the Cisco Unity IP Fax outbound process

Refer to the IP Fax Configuration Wizard and the Cisco Unity IP Fax Outbound Gateway Configuration Example.

- Basic knowledge of SMTP with Microsoft Exchange 2000/2003
- Cisco IOS Voice over IP (VoIP) configuration

An active Cisco Unity server with Cisco Unity IP Fax Configuration Wizard Version 2.0.0.19 or later is used in conjunction with the gateway configuration defined in this document in order to have a complete, operative system. On the IOS gateway side, a Cisco IOS router that supports T.37 OffRamp functionality and Cisco IOS Software Release 12.3(7)T or later is required along with Cisco Unity Version 4.0(4) or later with Microsoft Exchange 2000 or 2003 as the partner message store.

Note: T.37 OffRamp is not supported on Media Gateway Control Protocol (MGCP) networks. Refer to the T.37 Store and Forward Fax configuration guide for more information about platform and other restrictions when you use T.37.

Components Used

The information in this document is based on these software and hardware versions:

- Cisco 3725 as the IOS gateway
- Cisco IOS Software Release 12.3(8)T4

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Background Information

In order to illustrate the troubleshooting procedure, it is important to understand the basic operation of this feature. The problem is then narrowed down to a particular component and more information is gathered about it.
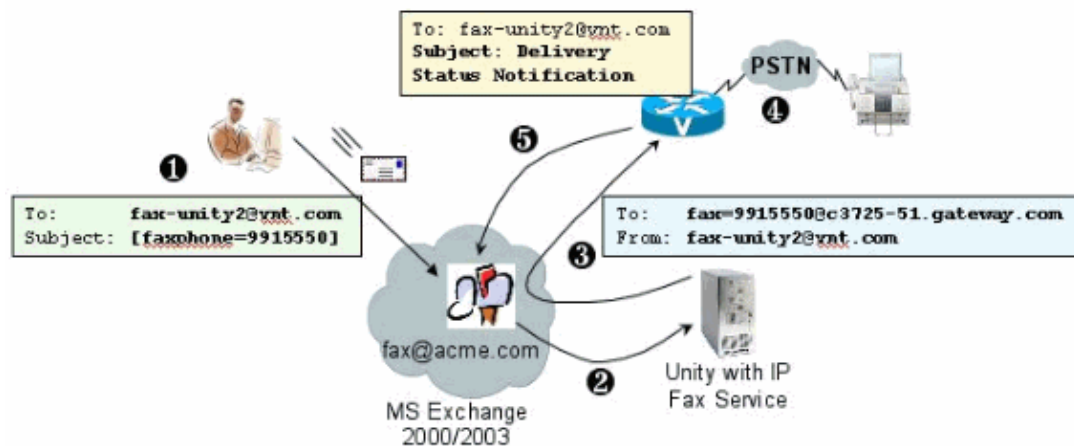
1. The end−user on the system sends an e−mail to the fax mailbox. This message must contain [FAXPHONE=9915550] in the subject to the fax mailbox (for example, fax−unity2@vnt.com), where

9915550 is the number of the destination fax machine. Based on the dial plan set up in the gateway, this number is either preceded with a 9 or only accepts certain digit strings. Once it is sent, Microsoft Exchange delivers the message to this mailbox.

Note: If the e−mail originates from another domain, then the Domain Name System (DNS) and the remote mail system are configured to deliver mail to the domain and mail account where the fax mailbox resides.

2. The Cisco Unity IP Fax service periodically logs into the fax mailbox and retrieves the message. Now the attachment is read. The message is rendered (for example, converted to a TIF file) on the Cisco Unity server, with the assumption that it is able to handle an attachment of that size.

3. Once the attachment is rendered, the message is sent to the IOS gateway. Microsoft Exchange must be configured to be able to send a message, which is addressed to a recipient, such as fax=9915550@c3725−51.gateway.com. This typically means that an A record and a mail exchange record (MX record) for that host (c3725−51.gateway.com) are added to DNS and an SMTP Connector is added to Exchange to route the message. The e−mail address and host name used here does not need to be user−friendly since end−users never send e−mail directly to the gateway.

4. The IOS gateway is configured for the T.37 OffRamp feature, which allows it to receive an SMTP message, interpret the fax=9915550 address, place a call to the 991550 number (after any possible digit manipulation), and then transmit the TIF file to the recipient fax machine.

Note: The gateway cannot use MGCP to use this feature. However, the circuit does not need to be dedicated to fax only. Voice and fax are used together on the same voice circuit. The dial−peer configuration on the router determines what is actually done with a call.

5. Once the fax is delivered, the gateway sends a Delivery Status Notification (DSN) message back to the fax mailbox (fax−unity2@vnt.com). This indicates whether the fax has been successfully delivered or not. When the IP Fax service logs in, the DSN message is retrieved. In the case where it has not been successfully delivered, the cause code is examined. Based on this cause code, either the message is resent (perhaps the remote fax was busy) or a message is sent back to the end−user to notify the user of a problem. Without this, the IP Fax service never knows if a fax was successfully delivered.

One of the advantages of this architecture is that any problem is usually broken down into the relevant components.

Before You Troubleshoot

Make Sure the CsIpFaxSvc Service is Installed, Configured, and Runs

Check the setting before you begin.

Set up the gateway properly. Refer to the Cisco Unity IP Fax Outbound Configuration Example and the IP Fax Configuration Wizard.

Set up Exchange so that the mail is delivered from the mailstore of the end−user to the one where the fax mailbox resides. It is delivered in the reverse direction also so that the status message can be sent back to the source. Mail needs to be able to deliver from the fax account to the gateway and back. This involves configuration in DNS and Exchange.

Ensure that the IP Fax service is properly installed. After the setup program is activated, run the configuration wizard on Cisco Unity and choose **Start > Programs > Unity > Cisco Unity IP Fax Configuration Wizard**.

The IP Fax service is not able to render anything you send to it. When the IP Fax service receives a file attached to an e−mail, it first checks to make sure that it can recognize the extension. These are set up in the

Third Party Fax Administration Tool. The default extensions allowed are BMP, HTM, TIF, and TXT. If a DOC extension is desired, then the tool has to add it as this example shows.

Note: There is an issue in some releases that prevents the last extension in the list of the Third Party Fax Administration Tool from proper operation. See the Caveats section for more information.
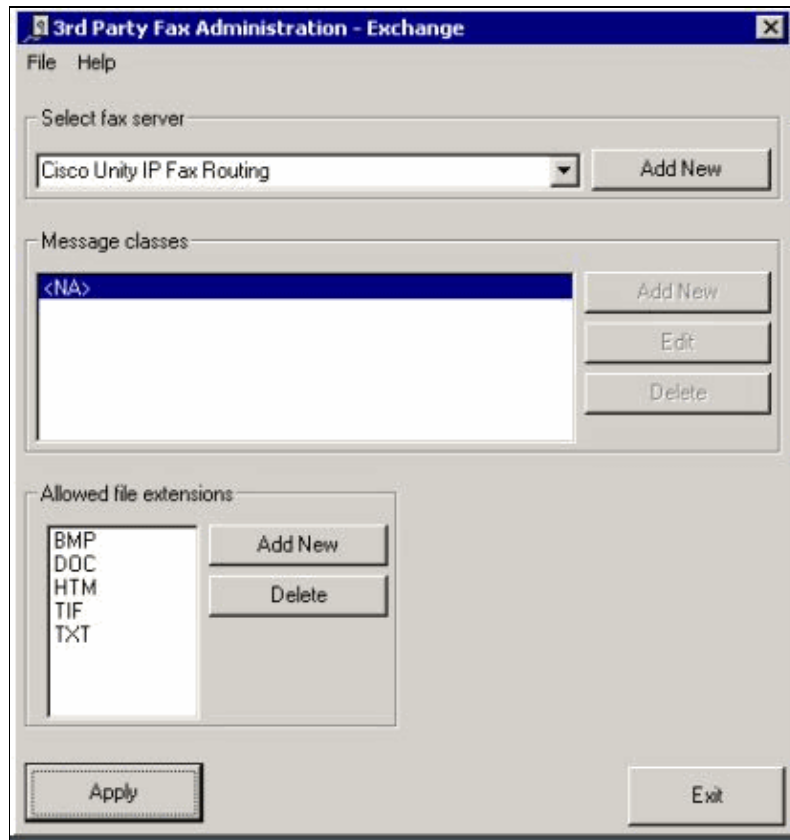
The fax server selected is Cisco Unity IP Fax Routing. DOC is added to the Allowed file extensions area. This means that the IP Fax service delivers the attachment to the rendering engine. Once this is done, the rendering engine relies on the application associated with the extension type so that it is able to open the document and print it to a file with its print driver. DOC extensions are typically allowed since the WordPad application is installed on the Cisco Unity server by default, and it opens most Microsoft Word documents. An application such as Acrobat Reader helps you to fax PDF files. See the Caveats section of this document for more information.

Note: The only application officially supported by Cisco for **.doc** extensions is WordPad. Rendering engine can fail if applications like Wordview or Microsoft Word is used with doc extension.

Note: Do not install Microsoft Office on the Cisco Unity server. This stops the proper operation of Cisco Unity. Only supported third−party applications should be installed on the Cisco Unity server. Be careful when you install any third−party software on the Cisco Unity server.

Verify from **Start > Administrative Tools > Services** that the CSIpFaxSvc is active and the Log On As field is the same as the AvCsMgr service. If the AvCsMgr service is not started, it is possible that there are problems. This also affects the CsIpFaxSvc. Refer to the Cisco Unity 3.1(x) and 4.x: AvCsMgr Service Fails to Start for more information.

Set Up a Mail Client to Monitor the Fax Account

If it does not work even after the IP Fax service and gateway are installed, the first step is to use Microsoft Outlook Express (with Internet Mail Access Protocol (IMAP) or Outlook (or another mail program) to examine the IP Fax mailbox. Since end–users send messages to this mailbox and the IP Fax service checks and sends messages from this account, make sure that the messages arrive and that they are sent properly. Everything seen here can be diagnosed with the use of the log files and the Exchange Message Tracking Center. But, this also helps to illustrate the operation of the product.

Outlook Express is installed automatically on the Cisco Unity server. Outlook is the mailer of choice with Unified Messaging deployments for Cisco Unity.

⚠ **Caution:** Do not install Microsoft Outlook on the Cisco Unity Server. It causes Cisco Unity to stop operation. Outlook should also not be installed on the Exchange server. It must be installed on a client machine and then configured in order to connect to the Exchange server.

When you add one or more mail accounts to Outlook Express, make sure that you specify the IMAP. If you select POP, it causes all mail messages to be downloaded from the mailstore immediately. Then the IP Fax service does not detect any new messages. Specify the IP Fax service account. If the password is not known, it is safe to access Active Directory Users and Computers, find the IP Fax account, and reset the password.

The troubleshooting method provided in this document is not the only one you can use. For instance, if you have access to the IP Fax account and are logged in through Outlook or Outlook Express, you can send a test e–mail directly from that location to an end–user. You can also send e–mails directly to the gateway. This narrows down the problem since it proves that one element or another works properly.

Registry Settings

In order to aid in troubleshooting, change the diagnostics level so that more output is logged. Choose **Start > Run** and enter **regedit** in the command box in order to enable this on the Cisco Unity server where the CSIPFaxSvc runs. Choose **HKEY_LOCAL_MACHINE\SOFTWARE\Active Voice\IPFaxRouter\1.0** once the Registry Editor tool appears.

Choose **Edit > New > DWORD Value**. The name specified is DiagLevel. When it is created, double–click this key. Enter **3** in the Value Data field. This is the maximum diagnostics level. At a later time, you can change it to 0 (the default). This change does not require you to restart the service.

⚠ **Caution:** When you change the wrong registry key or enter an incorrect value, it causes the server to malfunction. Confirm that you know how to restore it if a problem occurs before you edit the registry key. Refer to the Restoring topics within the Registry Editor Help for more information. A typical backup of the Cisco Unity server does not back up the registry. For a Cisco Unity failover system, registry changes on one Cisco Unity server are made manually on the other Cisco Unity server since registry changes are not replicated. Contact Cisco Technical Support for more information on how to change the registry key settings.
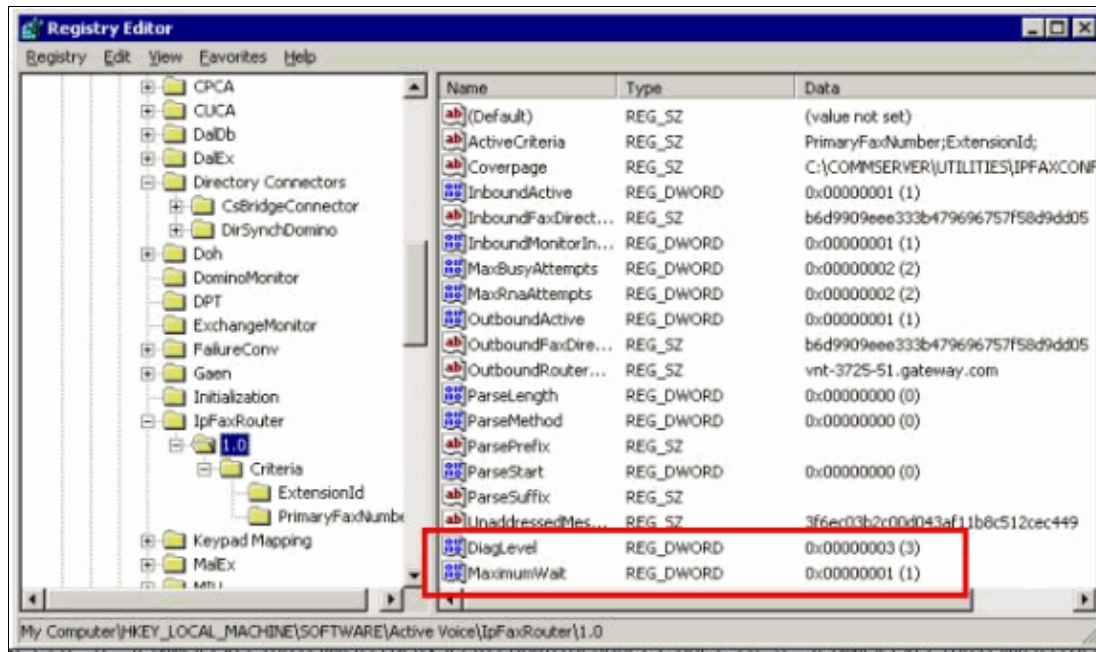
Another useful key in some situations is the MaximumWait key. If not present, this key defaults to 24 (hours). It specifies the maximum amount of time to hold a message before a non–delivery message is sent to the end–user. This key does not directly aid in troubleshooting, but it is frequently requested.

The OutboundRouterDomain key contains the e–mail domain that is used as the mail suffix when a mail message is addressed to an address to be delivered to the Cisco IOS gateway for outbound fax delivery. In order to do this, the value, vnt–3725–51.gateway.com in this example, must resolve through DNS. It can be helpful to ping this entry. If this entry is incorrect, an e–mail is returned to the sender and notifies them that the gateway entry cannot be resolved.

Tip: This fact is useful for troubleshooting: If you have everything installed except for the Cisco IOS gateway, you can temporarily change this registry entry to a bogus value, for example, something that is not in DNS. Then send a fax message to the fax mailbox. The CsIpFaxSvc service retrieves the message, renders the attachments, and then fails to locate the gateway. A Non–Delivery Receipt (NDR) notification is sent back to the sender with the rendered TIF attachment of the fax message that is sent to the gateway. If you do this, you have verified that:

- E–mail can be sent from your end–user to the fax account.
- Faxes can be rendered successfully.
- E–mail can be sent back from the fax account to the end–user.

Everything has been eliminated except for sending messages from the fax account to the gateway and from the gateway to the fax account.

The Coverpage key contains the location of the cover page that is rendered. Every fax that is transmitted contains a cover page with some information, such as the number of pages. It is possible to edit this page and to insert custom information or a company logo. If the key is changed to a non–existent file, then the cover page is not sent.

Troubleshoot Each Component

Step 1: Verify that Messages can be Sent from the End–User to the Fax Account

Send a message from a user to the fax account. In this diagram, the message is addressed to fax–unity2, which is the fax mailbox. The number that the user wants to call is specified in the subject line, which is 9915550 in this case. The document that the user wants to fax is attached.
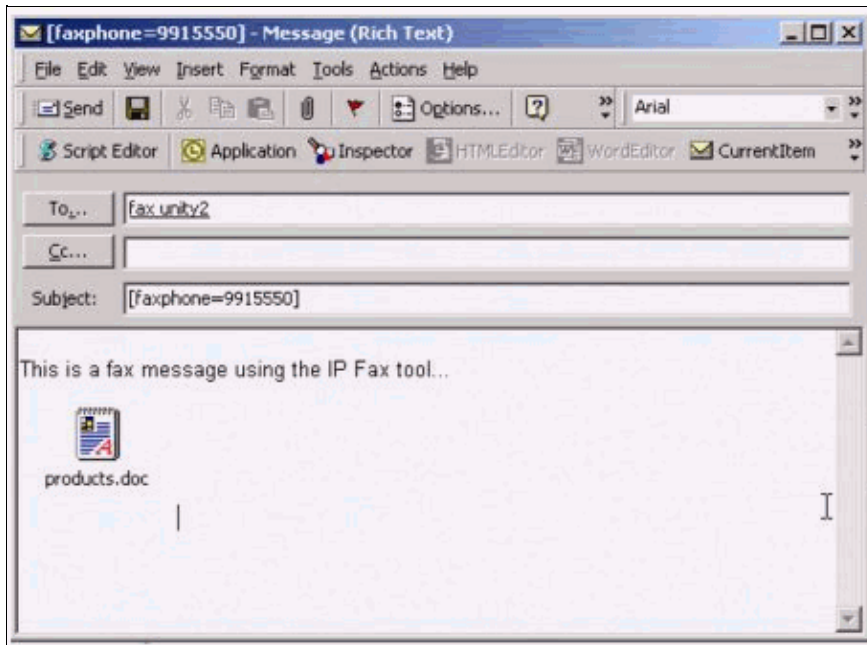
After this message is sent, it arrives in the fax mailbox that is set up initially. This figure shows the Outlook Express view of the mailbox.
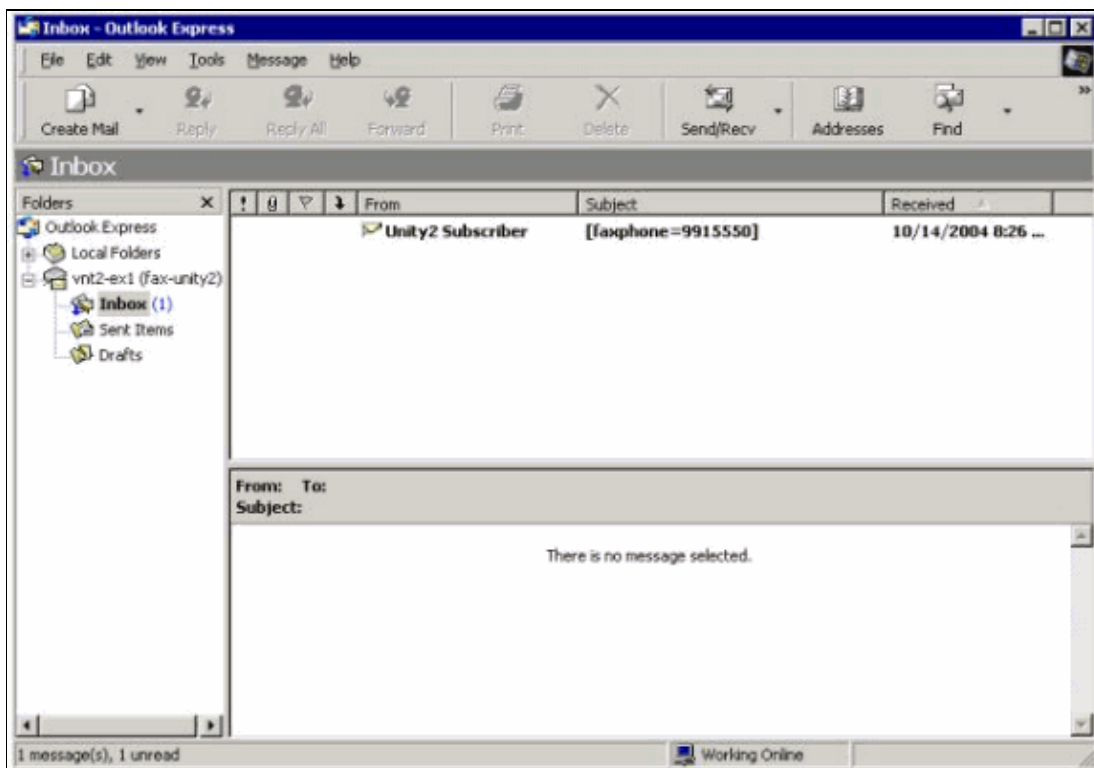
If the message does not arrive, there is a basic e–mail delivery issue. You should see if the sender received

some kind of a delivery failure message. Microsoft Exchange 2000/2003 has the Exchange Message Tracking feature, which allows you to see additional details about the mail delivery. If the sender and the fax account are in two separate domains, for example, a.com and b.com for instance, then make sure that the sending mail system is properly configured to deliver mail to the destination mail server. This involves setting up A and MX records in DNS and then adding an SMTP connector to route the mail. Mail should also be able to be sent in the reverse direction for the non−delivery messages to be sent back to the end−user.

This figure shows a DNS entry for the IOS gateway with A and MX records.
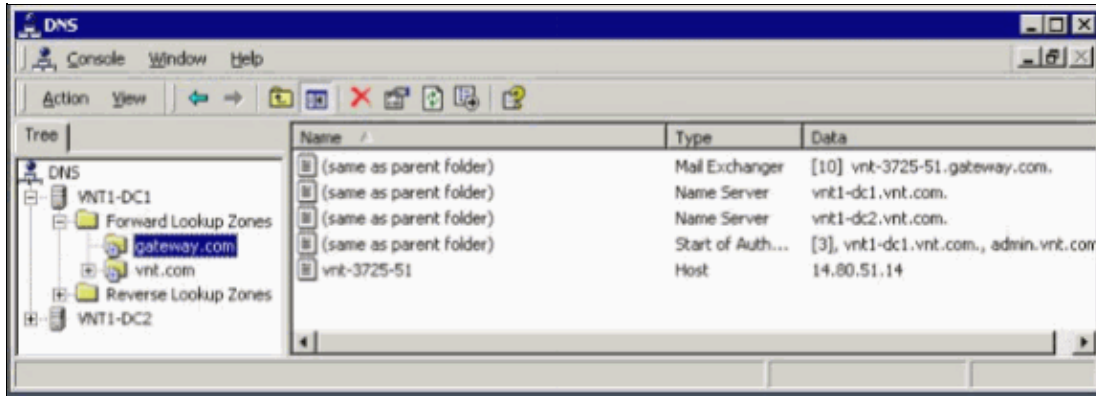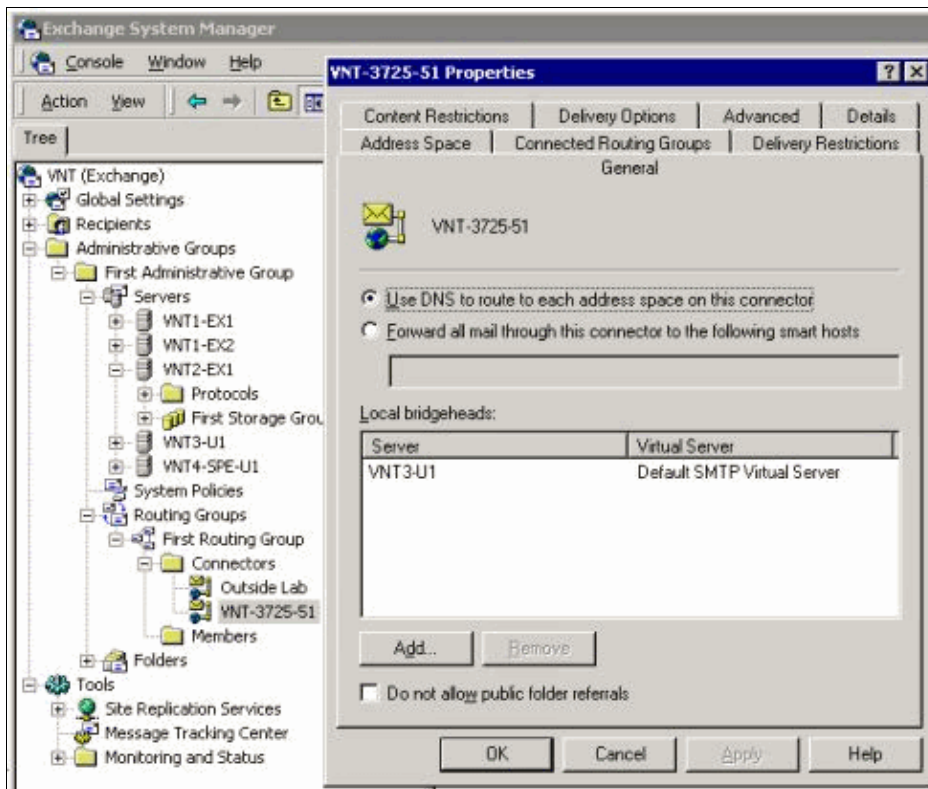
DNS is used to route to the address space. The address space is configured as c3725−51.gateway.com. This is only one of the many configurations. This is just the sample environment setup for a test. A typical corporate Exchange environment is much more complex. The point is to make sure that you get mail from the user to the fax mailbox. Check the Microsoft website for more information on these topics.

Make sure that the mail is sent from the end−user to the fax account and from the fax account to the ultimate end−users.

Step 2: Verify that Messages are Processed by the IP Fax Service

During the IP Fax service configuration, a polling interval is specified. This indicates how frequently the IP Fax service logs into the fax mailbox and checks for new messages. Cisco recommends setting this value low until the proper operation of the product is verified.

The best diagnostic log file is IPFaxSvc<date/time>.log. It is in the located in the \Commserver\logs directory.

Note: In this trace sample, some information has been removed.

You can see the mailbox that the log file logs into at each interval. You can also see it process the outbound faxes.

**Note:** All traces reflect the DiagLevel = 3 registry setting mentioned earlier.

```
"Logging on to mailbox [VNT2-EX1\fax-unity2]"
"CIpFaxCdoHelper::LogonToMailbox() name returned from Logon is [fax unity2]"
"Processing outbox, message count= 1"
"CIpFaxCdoHelper::ProcessFaxOutbox() processing message receipts"
"CIpFaxCdoHelper::DetermineMessageDisposition() current message
subject=[[faxphone=9915550]]"
"CIpFaxCdoHelper::DetermineMessageDisposition() setting disposition to OUTBOUND"
"CIpFaxCdoHelper::ProcessFaxOutbox() message looks like outbound fax - not
processing as receipt message"
```

It is important that the outbound process successfully logs into the correct mailbox, realizes that there is a message, retrieves the subject, and deems it to be an outbound fax.

```
"_RecordsetPtr::Open(Select * from tblIpFaxData where MessageEntryID =
'00000000377C58FB04A29C468DAB3CBC07D7F94B0700E7E65E238D0DEF46B5F8EAF983C6C512000000
05E5B20000E7E65E238D0DEF46B5F8EAF983C6C51200000005F55A0000') returned [0] rows -
Code: 0x80004005. Method: CIpFaxAdoHelper::GetRecordSet. [IpFaxAdoHelper.cpp::403]"

"CIpFaxAdoHelper::Execute() Cmd=[INSERT INTO tblIpFaxData (MessageEntryID, Status,
ScratchDirectory, TifFileName, NdrErrorCode, BusyCounter, RnaCounter,
TimeFirstHandled) VALUES ('00000000377C58FB04A29C468DAB3CBC07D7F94B0700E7E65E238D0
DEF46B5F8EAF983C6C51200000005E5B20000E7E65E238D0DEF46B5F8EAF983C6C51200000005F55A
0000', '0', 'C:\CommServer\unityMta\IpFaxScratchpad', '0188c93ff8194177b9f313edab
c2ea73.tif', '0', '0', '0', '2004-10-14 08:27:17')]"
```

Now the message is read. The first statement indicates that the service runs a SQL query against the tbllpFaxDatatable in the Reports database of the Cisco Unity server to see if a message with that MessageEntryID (the MAPI ID of the message) exists. It does not find any such entry (0 rows returned), so it adds a new row with this information. This MessageEntryID is used as the Subject field in the e–mail sent to the gateway. When you look at SQL directly you see:
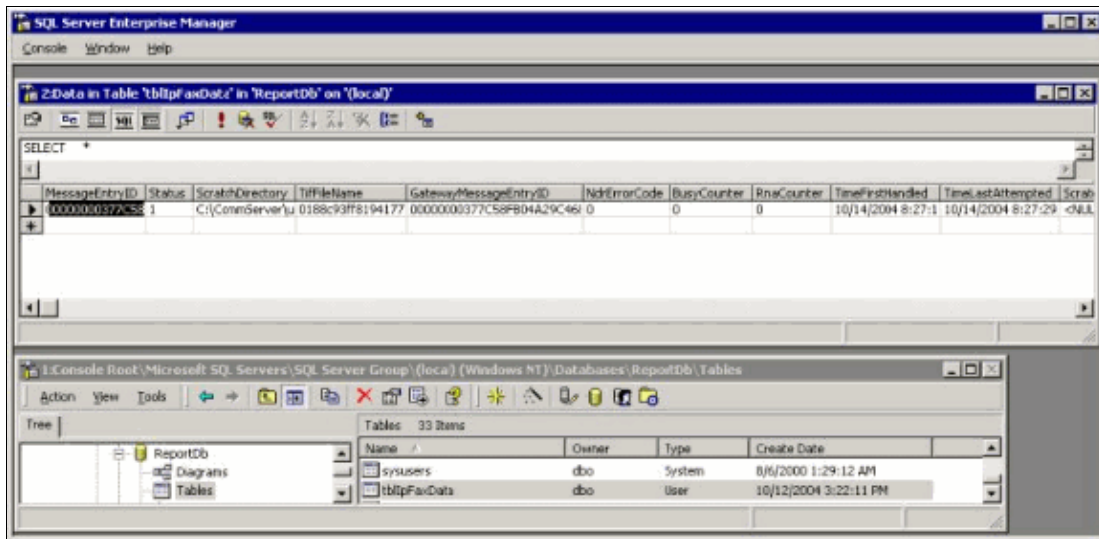
SQL is used to keep track of each message. When the fax mailbox is not empty, SQL goes through it at each interval and checks that it has accounted for each message based on this message ID. The attachment is copied to the scratch directory C:\CommServer\unityMta\IpFaxScratchpad in this case. If there is a DSN for this same message, it only needs to update the BusyCounter or something similar. But this is the method by which the service keeps track of each message. Now that the file has been copied to the scratch directory, the rendering takes place.
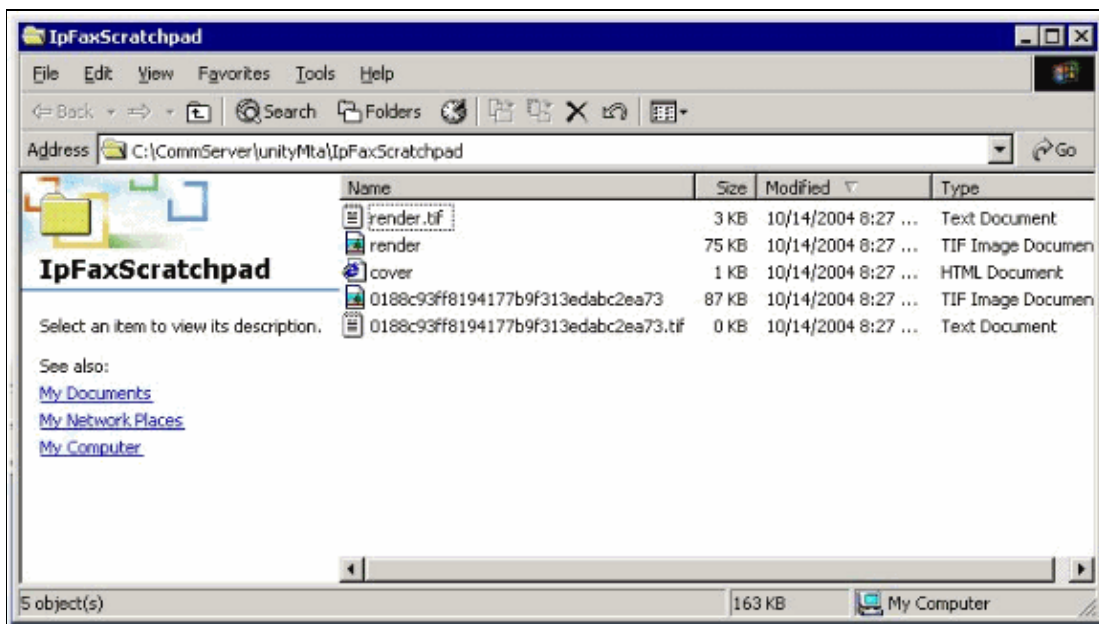

```
"CIpFaxCdoHelper::RenderCurrentMessage() scratch file=[C:\CommServer\
unityMta\IpFaxScratchpad\render.tif]"

"CIpFaxCdoHelper::RenderCurrentMessage() outputting message body to file
name=[C:\CommServer\unityMta\IpFaxScratchpad\MsgBody.txt]"

"Rendering file [C:\CommServer\unityMta\IpFaxScratchpad\MsgBody.txt] to
[C:\CommServer\unityMta\IpFaxScratchpad\render.tif]"

"CIpFaxCdoHelper::RenderCurrentMessage() added [1] pages to TIFF"
```

The rendering process starts. Every fax contains a cover page, a page with the message body, and any or all attachments. The cover page shows the number of pages that are faxed. The body is rendered first because it is always sent first. Since it only renders one message at a time, it always calls the message body MsgBody.txt

and the temporary output file render.tif. The rendering is performed by ImageMaker software. After each rendering process is completed, it returns the number of pages that are rendered. In this case, it is one page.

Step 3: Verify the Rendering Operation

The ImageMaker rendering engine is a print driver. It takes an input file, opens the associated default application (Wordpad, Notepad, and so forth), and uses the ImageMaker driver to print it to the file. This creates a TIF file.

The complete diagnostics files are in the Appendix section. When you submit a problem to Cisco Technical Support related to the rendering engine, these files are necessary:

- **C:\ImgMaker\Conversion_Toolkit\log\ODCDAEMON.log** The daemon that does the conversion through batch files
- **C:\log\ImageMakerDocCnvrt Driver.txt** The driver level information

The most useful information is in the ODCDAEMON.log file. For each file to be rendered, you see:

```
10/14/04 08:27:17
Start time: Thursday, October 14, 2004 08:27:17 AM
Daemon logged in as user: vnt2-u1-ms
Using registry root: HKEY_LOCAL_MACHINE
08:27:17: Pipe name: PrintDocDaemonPipe
08:27:17: Default printer name: ImageMaker DocCnvrt Driver
08:27:17: Registry server: <local>
08:27:17: Service started
08:27:19: 08:27:19: 10/14/04 08:27:19 ServiceRequest thread started
08:27:19: Processing 'Convert v1.0' request
08:27:19: Processing file: C:\CommServer\unityMta\IpFaxScratchpad\MsgBody.txt
08:27:19: Output file: C:\CommServer\unityMta\IpFaxScratchpad\render.tif
08:27:19: Printer name: ImageMaker DocCnvrt Driver
08:27:19: AppendTo filename set to C:\CommServer\unityMta\IpFaxScratchpad\render.tif
08:27:19: Input filename: C:\CommServer\unityMta\IpFaxScratchpad\MsgBody.txt
08:27:19: Output filename: C:\CommServer\unityMta\IpFaxScratchpad\render.tif
08:27:19: 0xE90: <10/14/04 08:27:19> StatusThread timeout value for jobstart: 60000
08:27:21: AutoShutdown feature activated
08:27:21: Sending response 1 to client
08:27:21: 10/14/04 08:27:21 ServiceRequest thread ended
```

The first element to look at is **logged in as user**. This is the same account that the CsIpFaxSvc service uses to log in.
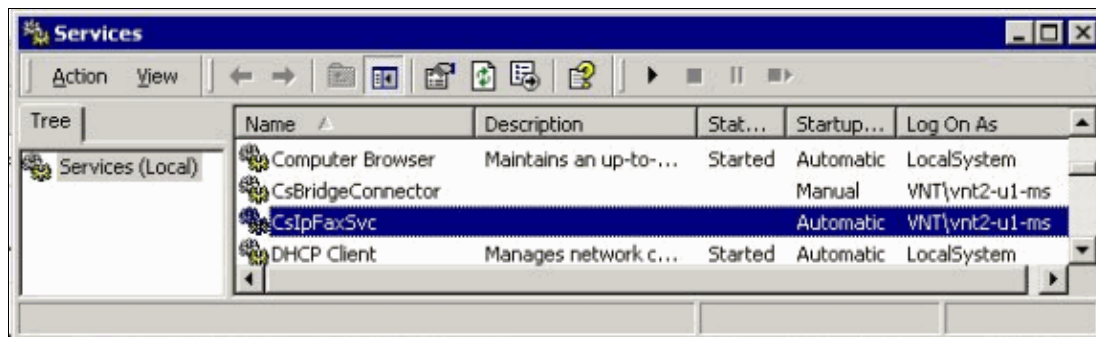
It is important to be aware of this because this user launches all applications and renders them. The next element is the **Input and output file names**. The third element, **response**, indicates the number of pages that are rendered. In this case it is one. If the response is negative, it indicates an error. If there is an error, you must look at the lines that immediately precede it. In many cases, the error occurs only for certain file types. The ODCDAEMON log file shows if the errors occur for all rendering or only for certain extension types.

It is beneficial to manually exercise the rendering engine to further test this. Log in as the same user that runs the CsIpFaxSvc service. This is to verify whether the same error occurs when the rendering engine is launched from the service.

If you have a file of each allowed extension type, open each file by double−clicking on it. The associated application opens and checks for updates (or a similar process). In the case of Acrobat, for example, this can be disabled.

Stop the CsIpFaxSvc. If any ODCDAEMON process still runs, stop it with the Task Manager.

Note: If the ODCDAEMON log file has to be saved for further reference, it has to be done before you ping the CsIpFaxSvc. When you restart the service, it clears the ODCDAEMON log file and the output is lost.
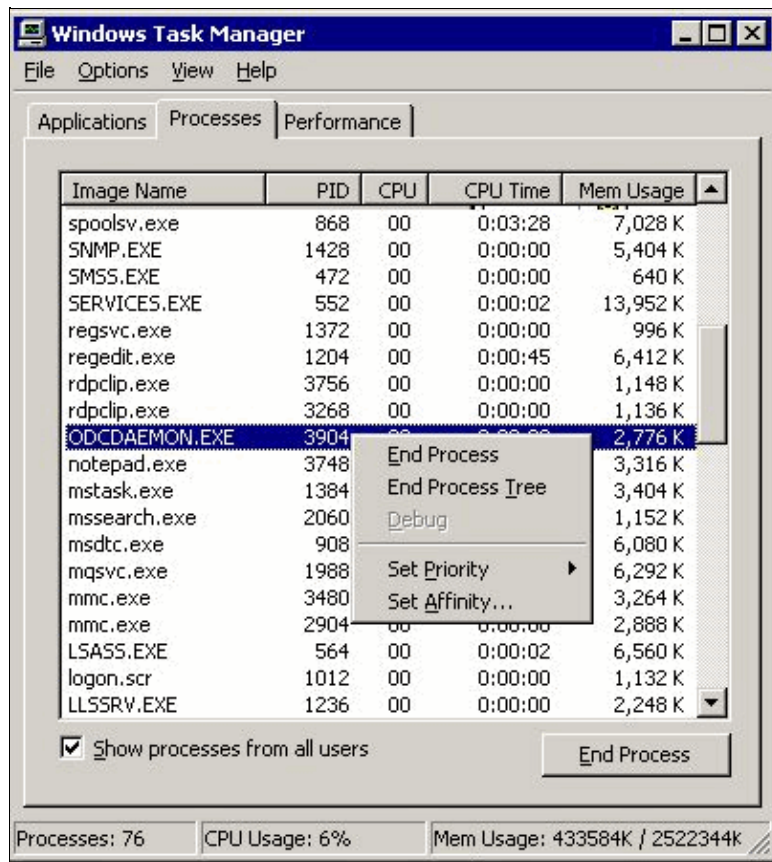
Launch the ODCDAEMON.EXE process from the folder C:\ImgMaker\Conversion_Toolkit. A window is displayed as this figure shows:
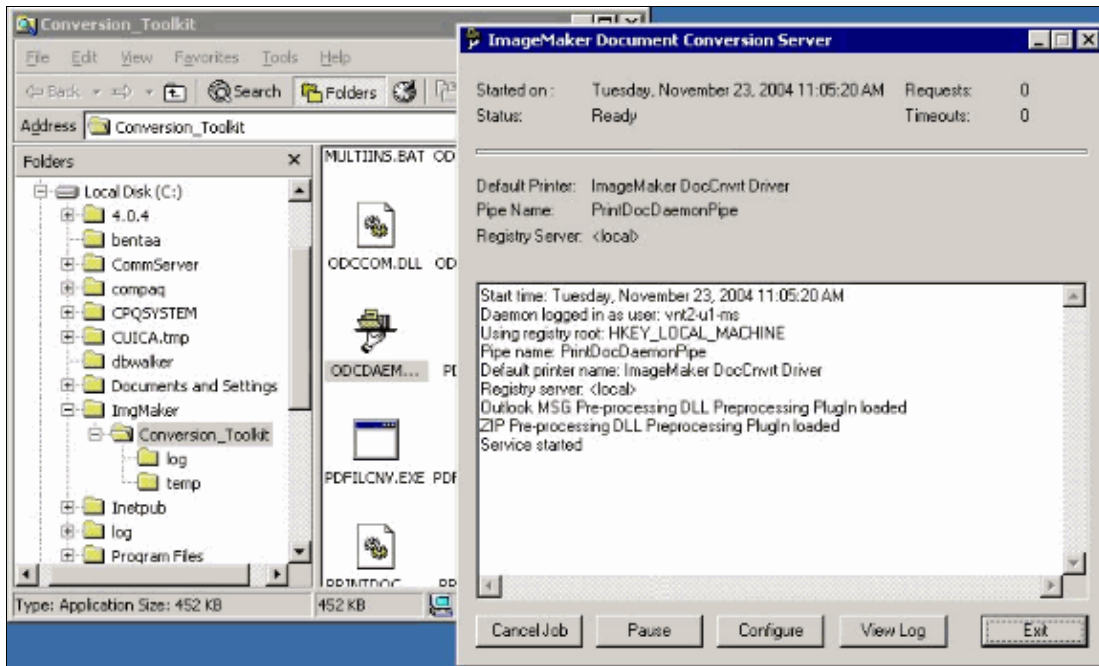
Launch the CPCOM.EXE file from the same folder.
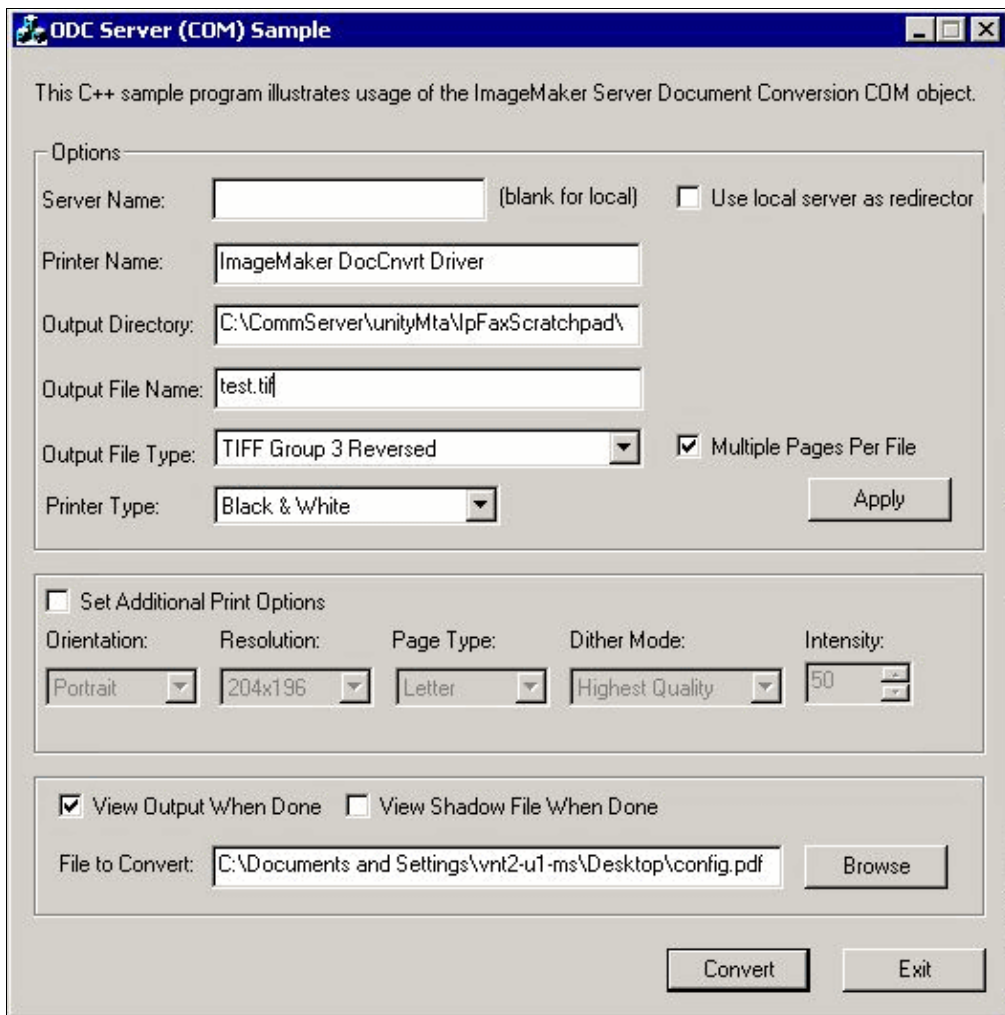
Most of the default settings are fine. In the File to Convert field, a desired input file is selected. Cisco recommends that you test every extension type separately. If the output file is not changed, the input file is

appended to the output file.

Click **Convert**. The application starts to print and this dialog box appears:

Click **OK**. Choose the **View Output When Done** option. A TIF viewer is launched and the output file displays. If there are errors, in some cases they are the same as what was presented in the log files as this example shows.

```
03:07:39: Input filename: d:\CommServer\UnityMTA\IpFaxScratchpad\render.tif
03:07:39: Output filename: d:\CommServer\UnityMTA\IpFaxScratchpad\9484f347996b4ea18b0caeb5
03:07:39: 0x15F0: <11/19/04 03:07:39> StatusThread timeout value for jobstart: 60000
PrintTo command does not exist for file type: MSPaper.Document
03:07:39: AutoShutdown feature activated
Conversion unsuccessful. Deleting file d:\CommServer\UnityMTA\IpFaxScratchpad\9484f347996b
03:07:39: Sending response -20 [PrintTo command does not exist] to client
```

The input file is a TIF attachment. But the file type is DOC (MSPaper.Document). The problem in this case is that Microsoft applications such as Word and Excel are installed on the server, and they change the file associations in the registry. Edit the HKEY_CLASSES_ROOT\.tif and TIF and change the Default key to TIFImage.Document instead of MSPaper.Document in order to fix this error.

Once each file type has been verified, close the CPCOM and ODCDAEMON windows and restart the CsIpFaxSvc service.

If all the attachments fail or there is an initialization error, reinstall the rendering engine. Use the **instfax.exe −z** command from the ImageMaker folder where the IPFaxConfigWizard resides in order to uninstall it. It is then reinstalled, which requires a reboot, and reruns the Cisco Unity IP Fax Configuration Wizard from **Start > Programs > Unity**.

The last option to test errors is to bypass the daemon process and render the file from the command line. In most cases, if it failed before, it also fails here. But, in some cases it provides additional information. This is done with the help of the PDFILCNV executable. This program is in the ImageMaker subdirectory of the IPFaxConfigWizard installation path. For example:

```
PDFILCNV products.doc -Fproducts.tif
```

Make sure that you can open the file that appears.

This example is from the IpFaxSvc log files:

```
"CIpFaxCdoHelper::RenderCurrentMessage() processing attachment=[products.doc]"

"CIpFaxCdoHelper::RenderCurrentMessage() attachment written to file=
[C:\CommServer\unityMta\IpFaxScratchpad\products.doc]"

"Rendering file [C:\CommServer\unityMta\IpFaxScratchpad\products.doc] to
[C:\CommServer\unityMta\IpFaxScratchpad\render.tif]"
```

```
"CIpFaxCdoHelper::R\

)ll
\-
\]
'==e-nderCurrentMessage() added [2] pages to TIFF"
```

The products DOC attachment is rendered. It is appended to the render.tif file created earlier. It has two pages, which makes the total number of pages three.

```
"CIpFaxCoverPage::GenerateFile() file=[C:\CommServer\unityMta\IpFaxScratchpad
\cover.htm] name=[Unity2 Subscriber] num=[9915550] pages=[3]"

"Rendering file [C:\CommServer\unityMta\IpFaxScratchpad\cover.htm] to
[C:\CommServer\unityMta\IpFaxScratchpad\0188c93ff8194177b9f313edabc2ea73.tif]"

"Rendering file [C:\CommServer\unityMta\IpFaxScratchpad\render.tif] to
[C:\CommServer\unityMta\IpFaxScratchpad\0188c93ff8194177b9f313edabc2ea73.tif]"
```

The cover page is the file cover.htm. It is a simple HTML template with some variables. It fills in the name of the subscriber who sends the message, the number called, and the number of pages sent.

After the custom cover.htm file is created, it is rendered into a messageid.tif file. The render.tif file created earlier is rendered, for example, appended, to the messageid.tif file. At this point the messageid.tif file is the real message that is e–mailed or faxed.

It is rendered successfully. Some fields in SQL are updated and the message is sent.

```
"2004-10-14","08:27:29","CIpFaxAdoHelper::Execute()
Cmd=[UPDATE tblIpFaxData SET Status = '1' WHERE MessageEntryID =
'00000000377C58FB04A29C468DAB3CBC07D7F94B0700E7E65E238D0DEF46B5F8EAF983C6C
51200000005E5B20000E7E65E238D0DEF46B5F8EAF983C6C51200000005F55A0000']"

"2004-10-14","08:27:29","CIpFaxAdoHelper::Execute() Cmd=[update tblipfaxdata
set TimeLastAttempted = '2004-10-14 08:27:29' where MessageEntryID =
'00000000377C58FB04A29C468DAB3CBC07D7F94B0700E7E65E238D0DEF46B5F8EAF983C6C5120
0000005E5B20000E7E65E238D0DEF46B5F8EAF983C6C51200000005F55A0000']";
```

The status of the message is set to 1, and the last updated time is filled in. It is sent to the gateway.

Step 4: Verify that the Router Receives and Delivers the Message

The message sent from the IP Fax account to the gateway must be in the form fax=####@router.gateway.com. Typically, DNS A and MX records are required in DNS and an SMTP connector needs to be added to Exchange to route messages to that host. Once this is done, you can proceed with troubleshooting the gateway.

If you only have the router set up, you can still verify that it works if you use Telnet in order to access the SMTP port and send a message manually as this example shows:

```
telnet vnt-3725-51.gateway.com 25
Trying 14.80.51.14...
Connected to 14.80.51.14.
Escape character is '^]'.
220 vnt-3725-51.gateway.com Cisco NetWorks ESMTP server
ehlo
250-vnt-3725-51.gateway.com, hello <empty HELO/EHLO> [172.18.106.123] (really )
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-PIPELINING
```

```
250-HELP
250-DSN
250 XACCOUNTING
mail from:test@test.com
250 2.5.0 Sender <test@test.com> ok
rcpt to:fax=9915550@vnt-3725-51.gateway.com
250 2.1.5 Recipient <fax=9915550@vnt-3725-51.gateway.com> ok, maps to `9915550' (cp=no)
data
354 Enter mail, end with a single "."
This is a test message from the router.
.
554 5.3.0 An unknown error occurred
quit
221 2.3.0 Goodbye from vnt-3725-51.gateway.com; closing connection
Connection closed by foreign host.
```

In this case there is an unknown error. This occurred because the Primary Rate Interface (PRI) that is used to deliver the outbound fax is down. This is another example:

```
telnet vnt-3725-51.gateway.com 25
Trying 14.80.51.14, 25 ... Open
220 vnt-3725-51.gateway.com Cisco NetWorks ESMTP server
ehlo
250-vnt-3725-51.gateway.com, hello <empty HELO/EHLO> [172.18.106.66] (really )
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-PIPELINING
250-HELP
250-DSN
250 XACCOUNTING
mail from:test@test.com
250 2.5.0 Sender <test@test.com> ok
rcpt to:fax=9915550@error.com
553 5.4.4 <fax=9915550@error.com> Unable to relay to remote host
quit
221 2.3.0 Goodbye from vnt-3725-51.gateway.com; closing connection
```

In this case the `Unable to relay to remote host` error message means that the `mta receive aliases` list in the Cisco IOS configuration does not have `error.com` configured.

Telnet to the router or access it through the console to begin troubleshooting the gateway directly. When you are Telnetted, issue the **terminal monitor** command. The **debug** output is seen. Turn on **debug fax mta all**. If the outbound voice circuit is a PRI, **debug isdn q931** is also a relevant command.

⚠ **Warning:** When you enable traces on a router, it sometimes affects the performance. Turn off the

traces with the **no debug all** command when it is finished.

Note: These logs have been truncated in order to save space.

This **debug** output is from an active example:

```
Oct 15 08:27:26.003: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
    context(0x66CA9AC4), state=1
Oct 15 08:27:35.051: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
    Calling smtp verb: ehlo
Oct 15 08:27:35.051: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
    context(0x66CA9AC4), state=1
Oct 15 08:27:35.599: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
    Calling smtp verb: mail
Oct 15 08:27:35.599: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
    context(0x66CA9AC4), state=1
```

```
Oct 15 08:27:36.143: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   Calling smtp verb: rcpt
Oct 15 08:27:36.143: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_command_rcpt:
   context(0x66CA9AC4)
Oct 15 08:27:36.143: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_command_rcpt:
   context(0x66CA9AC4)
Oct 15 08:27:36.143: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   context(0x66CA9AC4), state=1
Oct 15 08:27:36.691: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   Calling smtp verb: data
Oct 15 08:27:36.691: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   context(0x66CA9AC4), state=2
Oct 15 08:27:37.239: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   (S)R: `Received: from vnt2-ex1.vnt.com ([14.84.31.12]) by VNT1-EX2.vnt.com
with Microsoft SMTPSVC(5.0.2195.6713);'
Oct 15 08:27:37.239: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   (S)R: `        Thu, 14 Oct 2004 08:27:29 -0400'
Oct 15 08:27:37.239: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   (S)R: `Return-Receipt-To: "fax unity2"'
Oct 15 08:27:37.239: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   (S)R: `Subject: 00000000377C58FB04A29C468DAB3CBC07D7F94B0700E7E65E238D0DEF46B
5F8EAF983C6C51200000005E5B30000E7E65E238D0DEF46B5F8EAF983C6C51200000005FB680000'
Oct 15 08:27:37.239: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   (S)R: `Date: Thu, 14 Oct 2004 08:27:29 -0400'
Oct 15 08:27:37.239: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   (S)R: `Message-ID: '
Oct 15 08:27:37.239: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   (S)R: `X-MS-Has-Attach: yes'
Oct 15 08:27:37.239: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   (S)R: `MIME-Version: 1.0'
Oct 15 08:27:37.239: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   (S)R: `Content-Type: image/tiff;'
Oct 15 08:27:37.239: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   (S)R: `        name="BODY.TIF"'
Oct 15 08:27:37.239: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   (S)R: `X-MS-TNEF-Correlator: '
Oct 15 08:27:37.239: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   (S)R: `Content-Transfer-Encoding: base64'
Oct 15 08:27:37.239: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   (S)R: `Content-Description: BODY.TIF'
Oct 15 08:27:37.239: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   (S)R: `Content-Disposition: attachment;'
Oct 15 08:27:37.239: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   (S)R: `        filename="BODY.TIF"'
Oct 15 08:27:37.239: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   (S)R: `Thread-Topic: 00000000377C58FB04A29C468DAB3CBC07D7F94B0700E7E65E238D0
DEF46B5F8EAF983C6C51200000005E5B30000E7E65E238D0DEF46B5F8EAF983C6C51200000005
FB680000'
Oct 15 08:27:37.243: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   context(0x66CA9AC4), state=2
Oct 15 08:27:37.243: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   (S)R: `Thread-Index: AcSx6SvU9sc3nVffS3a4/Dwqi2Bpww=='
Oct 15 08:27:37.243: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   (S)R: `From: "fax unity2" '
Oct 15 08:27:37.243: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   (S)R: `content-class: urn:content-classes:message'
Oct 15 08:27:37.243: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   (S)R: `X-MimeOLE: Produced By Microsoft Exchange V6.0.5762.3'
Oct 15 08:27:37.243: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   (S)R: `To: '
Oct 15 08:27:37.243: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   (S)R: `Return-Path: fax-unity2@vnt.com'
Oct 15 08:27:37.243: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   (S)R: `X-OriginalArrivalTime: 14 Oct 2004 12:27:29.0487 (UTC) FILETIME=[2BE0DDF0:01C4B1
Oct 15 08:27:37.243: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_work_routine:
   (S)R: `'
```

```
Oct 15 08:27:37.243: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_new_part:
   Field name: `subject:'
   Field value: `00000000377C58FB04A29C468DAB3CBC07D7F94B0700E7E65E238D0DEF46B5F
8EAF983C6C51200000005E5B30000E7E65E238D0DEF46B5F8EAF983C6C51200000005FB680000'
Oct 15 08:27:37.243: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_new_part:
   Field name: `date:'
   Field value: `Thu, 14 Oct 2004 08:27:29 -0400'
Oct 15 08:27:37.243: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_new_part:
   Field name: `message-id:'
   Field value: `'
Oct 15 08:27:37.243: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_new_part:
   Field name: `Content-Type:'
   Field value: `image/tiff; name="BODY.TIF"'
Oct 15 08:27:37.243: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_new_part:
   Field name: `Content-Transfer-Encoding:'
   Field value: `base64'
Oct 15 08:27:37.243: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_new_part:
   Field name: `from:'
   Field value: `"fax unity2" '
Oct 15 08:27:37.243: //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_new_part:
   Field name: `to:'
   Field value: `'
Oct 15 08:27:37.247: //-1/E7B232638008/SMTPS/esmtp_server_engine_new_part:exit@1977
Oct 15 08:27:37.251: ISDN Se2/0:23 Q931: pak_private_number: Invalid type/plan
0x0 0x1 may be overriden; sw-type 13
Oct 15 08:27:37.251: ISDN Se2/0:23 Q931: Applying typeplan for sw-type 0xD is
0x0 0x0, Called num 99915550
Oct 15 08:27:37.255: ISDN Se2/0:23 Q931: TX -> SETUP pd = 8  callref = 0x0003
        Bearer Capability i = 0x8090A2
                Standard = CCITT
                Transer Capability = Speech
                Transfer Mode = Circuit
                Transfer Rate = 64 kbit/s
        Channel ID i = 0xA98397
                Exclusive, Channel 23
        Called Party Number i = 0x80, '99915550'
                Plan:Unknown, Type:Unknown
...
Oct 15 08:27:46.003: ISDN Se2/0:23 Q931: RX <- CONNECT pd = 8  callref = 0x8003
Oct 15 08:27:46.003: ISDN Se2/0:23 Q931: TX -> CONNECT_ACK pd = 8  callref = 0x0003
Oct 15 08:27:52.003: %ISDN-6-CONNECT: Interface Serial2/0:22 is now connected to
99915550
...
Oct 15 08:30:11.735: //-1/E7B232638008/SMTPS/esmtp_server_engine_buf_mgmt:
   buffer offset=1368, state=3
Oct 15 08:30:11.735: //-1/E7B232638008/SMTPS/esmtp_server_engine_dot_check:
   Not found the terminating Dot!
Oct 15 08:30:11.739: //-1/E7B232638008/SMTPS/esmtp_server_engine_buf_mgmt:
   state=3
Oct 15 08:30:11.739: //-1/E7B232638008/SMTPS/esmtp_server_engine_buf_mgmt:
   buffer offset=1412, state=3
Oct 15 08:30:11.739: //-1/E7B232638008/SMTPS/esmtp_server_engine_dot_check:
   Found the terminating Dot!
Oct 15 08:30:11.739: //-1/E7B232638008/SMTPS/esmtp_server_engine_work_routine:
context(0x66CA9AC4), state=10Oct 15 08:30:12.483: //-1/xxxxxxxxxxxx/SMTPS/
esmtp_server_engine_update_recipient_status:   faxmail status=6Oct 15 08:30:27.883:
 //-1/xxxxxxxxxxxx/SMTPS/esmtp_server_engine_update_recipient_status:
faxmail status=7
```

The mail server on the router receives the message. All the fields in the SMTP conversation get parsed at various levels of the router code until the call is made to the number. The important elements to notice are listed in a few configuration steps.

In the **show running–config** output, you see:

- **fax interface−type fax−mail**
- OffRamp software must be loaded into Flash or on an FTP/TFTP server. An application is then set up. For example, **call application voice offramp flash: app_faxmail_offramp.2.0.1.1.tcl**.

  Note: The Tool Command Language (TCL) software used for the OnRamp|OffRamp feature is downloaded from Cisco Downloads (registered customers only) .
- **mta receive maximum <recipients>** A number greater than 0
- **mta receive aliases <string>** Must match the DNS suffix. For example, if the e−mail from Cisco Unity is sent to fax=912345@myrouter.com then the alias must be myrouter.com. You can have multiple **mta receive aliases**.
- **mta send server <IP address of Exchange server>** In order to send DSNs, the gateway must connect to a mail server to deliver the message. Only one Exchange server needs to be configured, although multiple Exchange servers can be configured. Check whether all invalid **mta send server** addresses are removed. Otherwise, the first one in the list is always used.
- **mta receive generate permanent−error** This is required for the router to properly send DSNs back to Cisco Unity.
- Dial−peer configurations. For example:

```
dial-peer voice 5590 pots
      destination-pattern 991&.
      port 2/0:23
      forward-digits all
      prefix 9
 !
dial-peer voice 2 mmoip
   application offramp
   information-type fax
   incoming called-number 991
   dsn delayed
   dsn success
   dsn failure
 !
```

If the message does not arrive at the gateway, then you have to troubleshoot Exchange. The diagnostics within Exchange 2000 are very useful for this. For example:
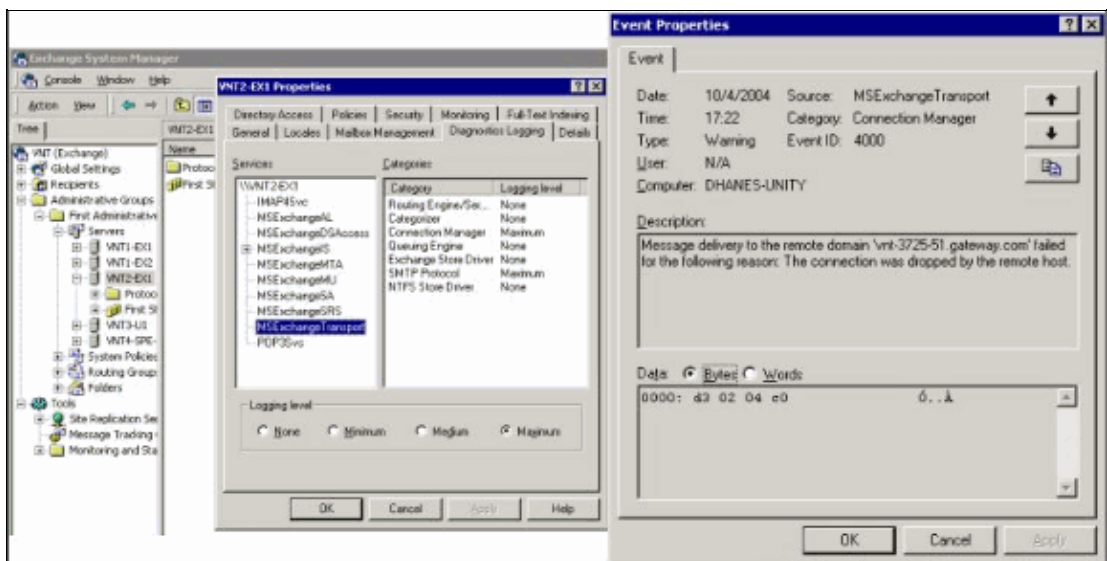
The MSExchangeTransport diagnostics of category SMTP Protocol and Connection Manager is enabled. This is in the Exchange System Manager under the individual server properties. When you send a message, the error is seen in the Application Event Viewer log. One of the most common misconfigurations is when the **mta receive aliases** parameter in the router does not match the e−mail

suffix. For example, if the fax service attempts to send a message to fax=91234@router.abc.com, then the gateway must have **mta receive aliases router.abc.com** configured.

Step 5: Verify that the Router Sends the DSN Message

This is the notification generated. Parts of the trace are omitted to save space.

```
Oct 15 08:30:41.739: //-1/E7B232638008/SMTPS/
esmtp_server_engine_generate_notification:
   context(0x66CA9AC4)
Oct 15 08:30:41.739: //-1/E7B232638008/SMTPS/sendDSN:
   context(0x66CA9AC4)
Oct 15 08:30:41.739: //-1/xxxxxxxxxxx/SMTPC/esmtp_client_engine_open:
   from=, to=fax-unity2@vnt.com
Oct 15 08:30:41.739: //-1/xxxxxxxxxxx/SMTPC/esmtp_client_engine_open:exit@1866
Oct 15 08:30:41.739: //-1/xxxxxxxxxxx/SMTPC/esmtp_client_engine_add_headers:
   from_comment=Fax Daemon
Oct 15 08:30:41.739: //-1/xxxxxxxxxxx/SMTPC/esmtp_client_engine_dispose_context:
exit@2109
Oct 15 08:30:41.739: //-1/E7B232638008/SMTPS/sendDSN:
   DSN sent, context(0x66CA9AC4)
Oct 15 08:30:41.739: //-1/E7B232638008/SMTPS/
esmtp_server_engine_generate_notification:exit@35
Oct 15 08:30:41.739: //-1/E7B232638008/SMTPS/esmtp_server_engine_work_routine:
   context(0x66CA9AC4), state=1
Oct 15 08:30:41.983: //-1/xxxxxxxxxxx/SMTPC/esmtp_client_engine_work_routine:
   socket 0 readable for first time
Oct 15 08:30:41.983: //-1/xxxxxxxxxxx/SMTPC/esmtp_client_engine_work_routine:
   socket 0 readable for first time
Oct 15 08:30:41.983: //-1/xxxxxxxxxxx/SMTPC/esmtp_client_engine_getln:
   (C)R: 220 vnt2-ex1.vnt.com Microsoft ESMTP MAIL Service, Version: 5.0.2195.6713
ready at  Thu, 14 Oct 2004 08:31:09 -0400
Oct 15 08:30:41.987: //-1/xxxxxxxxxxx/SMTPC/esmtp_client_engine_writeln:
   (C)S: EHLO vnt-3725-51.gateway.com
Oct 15 08:30:41.987: //-1/xxxxxxxxxxx/SMTPC/esmtp_client_engine_write:
   return code=0
Oct 15 08:30:42.463: //-1/xxxxxxxxxxx/SMTPC/esmtp_client_engine_getln:
   (C)R: 250-vnt2-ex1.vnt.com Hello [14.80.51.14]
Oct 15 08:30:42.463: //-1/xxxxxxxxxxx/SMTPC/esmtp_client_engine_getln:
   (C)R: 250-TURN
Oct 15 08:30:42.463: //-1/xxxxxxxxxxx/SMTPC/esmtp_client_engine_getln:
   (C)R: 250-ATRN
Oct 15 08:30:42.463: //-1/xxxxxxxxxxx/SMTPC/esmtp_client_engine_getln:
   (C)R: 250-SIZE
Oct 15 08:30:42.463: //-1/xxxxxxxxxxx/SMTPC/esmtp_client_engine_getln:
   (C)R: 250-ETRN
Oct 15 08:30:42.463: //-1/xxxxxxxxxxx/SMTPC/esmtp_client_engine_getln:
   (C)R: 250-PIPELINING
Oct 15 08:30:42.463: //-1/xxxxxxxxxxx/SMTPC/esmtp_client_engine_getln:
   (C)R: 250-DSN
Oct 15 08:30:42.463: //-1/xxxxxxxxxxx/SMTPC/esmtp_client_engine_getln:
   (C)R: 250-ENHANCEDSTATUSCODES
Oct 15 08:30:42.463: //-1/xxxxxxxxxxx/SMTPC/esmtp_client_engine_getln:
   (C)R: 250-8bitmime
Oct 15 08:30:42.463: //-1/xxxxxxxxxxx/SMTPC/esmtp_client_engine_getln:
   (C)R: 250-BINARYMIME
Oct 15 08:30:42.463: //-1/xxxxxxxxxxx/SMTPC/esmtp_client_engine_getln:
   (C)R: 250-CHUNKING
Oct 15 08:30:42.463: //-1/xxxxxxxxxxx/SMTPC/esmtp_client_engine_getln:
   (C)R: 250-VRFY
Oct 15 08:30:42.463: //-1/xxxxxxxxxxx/SMTPC/esmtp_client_engine_getln:
   (C)R: 250-X-EXPS GSSAPI NTLM LOGIN
Oct 15 08:30:42.463: //-1/xxxxxxxxxxx/SMTPC/esmtp_client_engine_getln:
   (C)R: 250-X-EXPS=LOGIN
```

```
Oct 15 08:30:42.463: //-1/xxxxxxxxxxxx/SMTPC/esmtp_client_engine_getln:
   (C)R: 250-AUTH GSSAPI NTLM LOGIN
Oct 15 08:30:42.463: //-1/xxxxxxxxxxxx/SMTPC/esmtp_client_engine_getln:
   (C)R: 250-AUTH=LOGIN
Oct 15 08:30:42.463: //-1/xxxxxxxxxxxx/SMTPC/esmtp_client_engine_getln:
   (C)R: 250-X-LINK2STATE
Oct 15 08:30:42.467: //-1/xxxxxxxxxxxx/SMTPC/esmtp_client_engine_getln:
   (C)R: 250-XEXCH50
Oct 15 08:30:42.467: //-1/xxxxxxxxxxxx/SMTPC/esmtp_client_engine_getln:
   (C)R: 250 OK
Oct 15 08:30:42.467: //-1/xxxxxxxxxxxx/SMTPC/esmtp_client_engine_writeln:
   (C)S: MAIL FROM:<>
Oct 15 08:30:42.467: //-1/xxxxxxxxxxxx/SMTPC/esmtp_client_engine_write:
   return code=0
Oct 15 08:30:43.011: //-1/xxxxxxxxxxxx/SMTPC/esmtp_client_engine_getln:
   (C)R: 250 2.1.0 <>....Sender OK
Oct 15 08:30:43.011: //-1/xxxxxxxxxxxx/SMTPC/esmtp_client_engine_writeln:
   (C)S: RCPT TO:
Oct 15 08:30:43.011: //-1/xxxxxxxxxxxx/SMTPC/esmtp_client_engine_write:
   return code=0
Oct 15 08:30:43.555: //-1/xxxxxxxxxxxx/SMTPC/esmtp_client_engine_getln:
   (C)R: 250 2.1.5 fax-unity2@vnt.com
Oct 15 08:30:43.555: //-1/xxxxxxxxxxxx/SMTPC/esmtp_client_engine_write:
   return code=0
Oct 15 08:30:44.103: //-1/xxxxxxxxxxxx/SMTPC/esmtp_client_engine_getln:
   (C)R: 354 Start mail input; end with .
Oct 15 08:30:44.103: //-1/xxxxxxxxxxxx/SMTPC/esmtp_client_engine_writeln:
   (C)S: Received:  by vnt-3725-51.gateway.com for  (with Cisco NetWorks);
Fri, 15 Oct 2004 08:30:44 +0000
Oct 15 08:30:44.107: //-1/xxxxxxxxxxxx/SMTPC/esmtp_client_engine_write:
   return code=0
Oct 15 08:30:44.107: //-1/xxxxxxxxxxxx/SMTPC/esmtp_client_engine_writeln:
   (C)S: To:
Oct 15 08:30:44.107: //-1/xxxxxxxxxxxx/SMTPC/esmtp_client_engine_write:
   return code=0
Oct 15 08:30:44.107: //-1/xxxxxxxxxxxx/SMTPC/esmtp_client_engine_writeln:
   (C)S: Message-ID: <00042004083044107@vnt-3725-51.gateway.com>
Oct 15 08:30:44.107: //-1/xxxxxxxxxxxx/SMTPC/esmtp_client_engine_write:
   return code=0
Oct 15 08:30:44.107: //-1/xxxxxxxxxxxx/SMTPC/esmtp_client_engine_writeln:
   (C)S: Date: Fri, 15 Oct 2004 08:30:44 +0000
Oct 15 08:30:44.107: //-1/xxxxxxxxxxxx/SMTPC/esmtp_client_engine_write:
   return code=0
Oct 15 08:30:44.107: //-1/xxxxxxxxxxxx/SMTPC/esmtp_client_engine_writeln:
   (C)S: Subject: Delivery Status Notification
...
```

If it does not work, the **mta send server** command in the configuration must be set to the correct Exchange server and port (25 by default). DSN needs to be configured on the mmoip dial–peer that the fax message matches when it is sent, and the **mta receive generate permanent–error** command is present. Sometimes the command is not present, or it points to an incorrect mail server.

Note: The Cisco IOS configuration allows you to configure multiple **mta send server** commands. But, only the first one is used, unless that device becomes completely unreachable. So, when you change the **mta send server**, you have to remove the previous one.

If you have Outlook or Outlook Express and you monitor the fax mailbox, you see the DSN get delivered there before it is processed in the next interval by the IP Fax service.
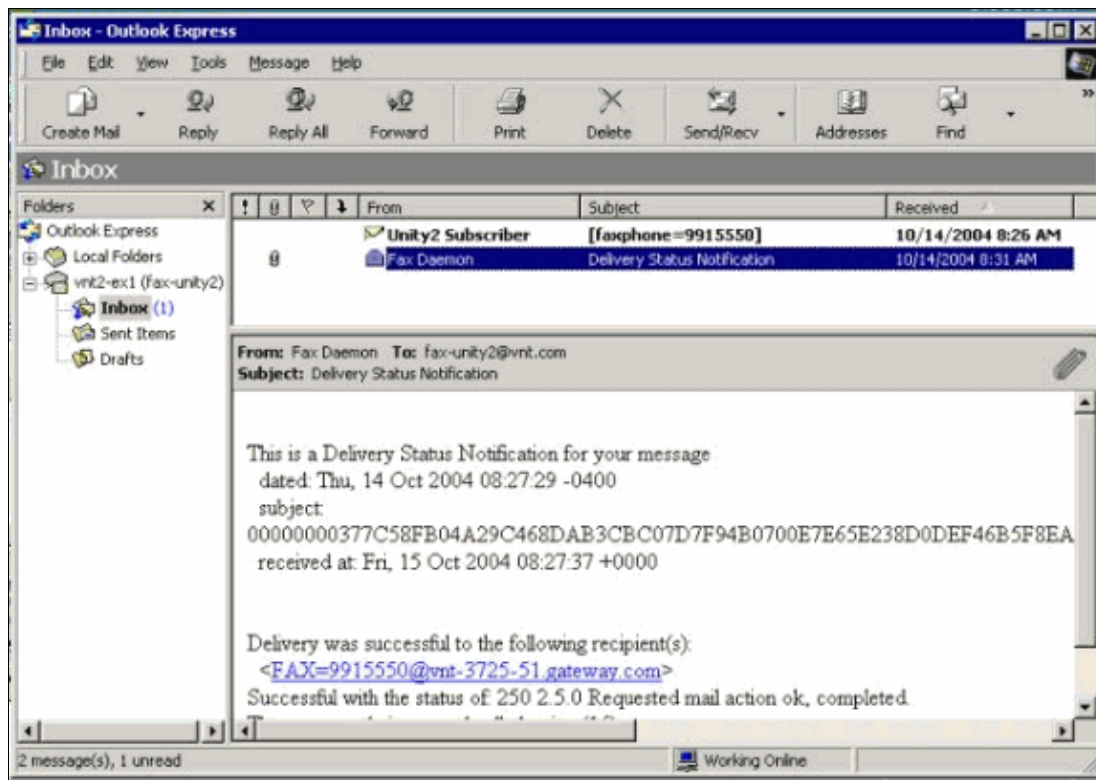
The body of the message contains the message ID. Previously it was the subject. Errors are mentioned near the bottom of the message body. They are abstractions of the ISDN Q.931 error codes. See the Appendix for more information.

An equally useful tool is the Exchange Message Tracking Center. It is built into the Exchange 2000/2003 System Manager. These are the mail messages:
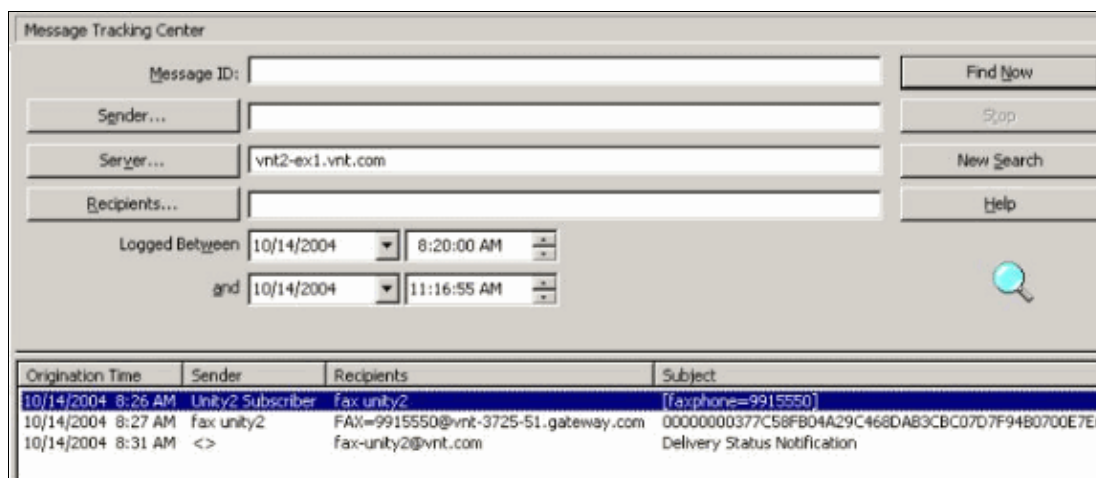
Step 6: Verify that the DSN Message was Received

When the DSN message is in the mailbox of the IP Fax service, check in order to make sure that it processes the message. The IpFaxSvc log files are helpful with this.

```
"2004-10-14","08:31:30","Logging on to mailbox [VNT2-EX1\fax-unity2]"
"2004-10-14","08:31:30","CIpFaxCdoHelper::LogonToMailbox() name returned from
Logon is [fax unity2]""2004-10-14","08:31:30","Processing outbox, message count= 2"
"2004-10-14","08:31:30","CIpFaxCdoHelper::ProcessFaxOutbox() processing message
receipts"
```

```
"2004-10-14","08:31:30","CIpFaxCdoHelper::DetermineMessageDisposition()
current message subject=[Delivery Status Notification]"
"2004-10-14","08:31:30","CIpFaxCdoHelper::MessageIsReceipt() processing message,
type=[IPM.Note]""2004-10-14","08:31:30","CIpFaxCdoHelper::MessageIsReceipt()
processing message as DELIVERY STATUS NOTIFICATION"
```

The message is processed.

```
"2004-10-14","08:31:30","CIpFaxMessageBodyParser::GetOriginalSubject
returning subject=[00000000377C58FB04A29C468DAB3CBC07D7F94B0700E7E65E238D0DEF4
6B5F8EAF983C6C51200000005E5B30000E7E65E238D0DEF46B5F8EAF983C6C51200000005FB680000]"
"2004-10-14","08:31:30","CIpFaxMessageBodyParser::GetCauseCode returning cause
code =[16]"
```

The Original Subject is parsed from the body of the message. This allows you to look the message up in the database. The Cause Code is also parsed as 16. This is a normal call clearing. These cause codes correspond to Q.931 cause codes. See Q.931 Disconnect Cause Codes for more information on cause codes.

The entry is safely removed from the database and re−added with Status 0.

```
"2004-10-14","08:31:30","CIpFaxAdoHelper::Execute() Cmd=[DELETE
FROM tblIpFaxData WHERE MessageEntryID = '00000000377C58FB04A29C468DAB3CBC07D
7F94B0700E7E65E238D0DEF46B5F8EAF983C6C51200000005E5B20000E7E65E238D0DEF46B5F8
EAF983C6C51200000005F55A0000']"

"2004-10-14","08:31:30","_RecordsetPtr::Open(Select * from tblIpFaxData where
MessageEntryID = '00000000377C58FB04A29C468DAB3CBC07D7F94B0700E7E65E238D0DEF46
B5F8EAF983C6C51200000005E5B20000E7E65E238D0DEF46B5F8EAF983C6C51200000005F55A0000')
 returned [0] rows − Code: 0x80004005. Method: CIpFaxAdoHelper::GetRecordSet.
[IpFaxAdoHelper.cpp::403]"

"2004-10-14","08:31:30","CIpFaxAdoHelper::Execute() Cmd=[INSERT INTO tblIpFaxData
 (MessageEntryID, Status, ScratchDirectory, TifFileName, NdrErrorCode,
BusyCounter, RnaCounter, TimeFirstHandled) VALUES ('00000000377C58FB04A29C4
68DAB3CBC07D7F94B0700E7E65E238D0DEF46B5F8EAF983C6C51200000005E5B20000E7E65E
238D0DEF46B5F8EAF983C6C51200000005F55A0000', '0', 'C:\CommServer\unityMta\
IpFaxScratchpad', '2fd36f2af94d4137835928894fb397eb.tif', '0', '0', '0',
'2004-10-14 08:31:30')]"
```

Both mail messages are deleted .When you check for outbound fax messages, there are none.

```
2004-10-14","08:32:30","Processing outbox, message count= 0"
```

Caveats

These are a few current caveats that are present in the currently tested version. Refer to the IP Fax Configuration Wizard or the Help file for more current releases that address these issues.

- Cisco bug ID CSCef61241 (registered customers only) Outbound IP Fax can not render PDFs.
- Cisco bug ID CSCef61256 (registered customers only) Outbound IP Fax can not handle gateway DSNs with Exchange 5.5.
- Cisco bug ID CSCef96036 (registered customers only) IP Fax service should send NDR when subject incorrectly formatted.
- Cisco bug ID CSCef96051 (registered customers only) Outbound IP Fax service cover page should allow a callback number.
- Cisco bug ID CSCeg04158 (registered customers only) IP Fax service should NDR a message with an unknown attachment.
- Cisco bug ID CSCeg04180 (registered customers only) IP Fax service does not recognize the last extension in the third−party fax tool.

- Cisco bug ID CSCeg39033 (registered customers only) IP Fax service should allow you to not render the e−mail message body.
- Cisco bug ID CSCeg39046 (registered customers only) IP Fax rendering engine and log directory is always on the C: drive.
- Cisco bug ID CSCeg39100 (registered customers only) IP Fax service should NDR a message after multiple rendering failures.
- Cisco bug ID CSCeg45439 (registered customers only) IP Fax service should allow confirmation of fax delivery to sender.
- Cisco bug ID CSCeg45666 (registered customers only) IP Fax service should check for application dependencies on startup.
- Cisco bug ID CSCeg46458 (registered customers only) Stopping IP Fax service should kill the ODCDAEMON process.
- Cisco bug ID CSCeg52736 (registered customers only) In the outbound IP Fax service, the MaximumWait key does not work.

Appendix

Render Engine Trace Files

These are traces of the example discussed in this document. There are two files. One file is for the daemon and the other file is for the print driver. Typically these are not needed. But, a sample of each is included for comparison purposes. When you submit problems with the rendering engine, these files are required along with the regular IP Fax logs.

ODCDAEMON.log

```
C:\ImgMaker\Conversion_Toolkit\log\ODCDAEMON.log
```

This file is used on a frequent basis and is fairly legible. The input and output files are listed in a separate section and a response code is listed. If the response code is positive, then it indicates the number of pages that are rendered into the output TIF file. A negative response code indicates an error. Check the user that the process runs as. This is critical, since this is the user that must be able to open the individual applications to convert them to TIF.

```
10/14/04 08:27:17
Start time: Thursday, October 14, 2004 08:27:17 AM
Daemon logged in as user: vnt2-u1-ms
Using registry root: HKEY_LOCAL_MACHINE
08:27:17: Pipe name: PrintDocDaemonPipe
08:27:17: Default printer name: ImageMaker DocCnvrt Driver
08:27:17: Registry server: <local>
08:27:17: Service started
08:27:19:
08:27:19: 10/14/04 08:27:19 ServiceRequest thread started
08:27:19: Processing 'Convert v1.0' request
08:27:19: Processing file: C:\CommServer\unityMta\IpFaxScratchpad\MsgBody.txt
08:27:19: Output file: C:\CommServer\unityMta\IpFaxScratchpad\render.tif
08:27:19: Printer name: ImageMaker DocCnvrt Driver
08:27:19: AppendTo filename set to C:\CommServer\unityMta\IpFaxScratchpad\render.tif
08:27:19: Input filename: C:\CommServer\unityMta\IpFaxScratchpad\MsgBody.txt
08:27:19: Output filename: C:\CommServer\unityMta\IpFaxScratchpad\render.tif
08:27:19: 0xE90: <10/14/04 08:27:19> StatusThread timeout value for jobstart: 60000
08:27:21: AutoShutdown feature activated
08:27:21: Sending response 1 to client
08:27:21: 10/14/04 08:27:21 ServiceRequest thread ended
08:27:21:
08:27:21: 10/14/04 08:27:21 ServiceRequest thread started
08:27:21: Processing 'Convert v1.0' request
```

```
08:27:21: Processing file: C:\CommServer\unityMta\IpFaxScratchpad\products.doc
08:27:21: Output file: C:\CommServer\unityMta\IpFaxScratchpad\render.tif
08:27:21: Printer name: ImageMaker DocCnvrt Driver
08:27:21: AppendTo filename set to C:\CommServer\unityMta\IpFaxScratchpad\render.tif
08:27:21: Input filename: C:\CommServer\unityMta\IpFaxScratchpad\products.doc
08:27:21: Output filename: C:\CommServer\unityMta\IpFaxScratchpad\render.tif
08:27:21: 0xEB8: <10/14/04 08:27:21> StatusThread timeout value for jobstart: 60000
08:27:23: AutoShutdown feature activated
08:27:23: Sending response 2 to client
08:27:23: 10/14/04 08:27:23 ServiceRequest thread ended
08:27:23:
08:27:23: 10/14/04 08:27:23 ServiceRequest thread started
08:27:23: Processing 'Convert v1.0' request
08:27:23: Processing file: C:\CommServer\unityMta\IpFaxScratchpad\cover.htm
08:27:23: Output file: C:\CommServer\unityMta\IpFaxScratchpad\
           0188c93ff8194177b9f313edabc2ea73.tif
08:27:23: Printer name: ImageMaker DocCnvrt Driver
08:27:23: AppendTo filename set to C:\CommServer\unityMta\
           IpFaxScratchpad\0188c93ff8194177b9f313edabc2ea73.tif
08:27:23: Input filename: C:\CommServer\unityMta\IpFaxScratchpad\cover.htm
08:27:23: Output filename: C:\CommServer\unityMta\IpFaxScratchpad\
           0188c93ff8194177b9f313edabc2ea73.tif
08:27:23: 0xED8: <10/14/04 08:27:23> StatusThread timeout value for jobstart: 60000
08:27:24: AutoClose skipping window: <Print> with class: <#32770> because parent is IE6
08:27:24: AutoClose skipping window: <Print> with class: <#32770> because parent is IE6
08:27:26: AutoShutdown feature activated
08:27:26: Sending response 1 to client
08:27:26: 10/14/04 08:27:26 ServiceRequest thread ended
08:27:26:
08:27:26: 10/14/04 08:27:26 ServiceRequest thread started
08:27:26: Processing 'Convert v1.0' request
08:27:26: Processing file: C:\CommServer\unityMta\IpFaxScratchpad\render.tif
08:27:26: Output file: C:\CommServer\unityMta\IpFaxScratchpad\
           0188c93ff8194177b9f313edabc2ea73.tif
08:27:26: Printer name: ImageMaker DocCnvrt Driver
08:27:26: AppendTo filename set to C:\CommServer\unityMta\IpFaxScratchpad\
           0188c93ff8194177b9f313edabc2ea73.tif
08:27:26: Input filename: C:\CommServer\unityMta\IpFaxScratchpad\render.tif
08:27:26: Output filename: C:\CommServer\unityMta\IpFaxScratchpad\
           0188c93ff8194177b9f313edabc2ea73.tif
08:27:26: 0xEDC: <10/14/04 08:27:26> StatusThread timeout value for jobstart: 60000
08:27:29: AutoShutdown feature activated
08:27:29: Sending response 3 to client
08:27:29: 10/14/04 08:27:29 ServiceRequest thread ended
```

Driver Log

```
C:\log\ImageMaker DocCnvrt Driver.log
```

This trace is necessary when you submit problems with the rendering engine to Cisco Technical Support.

```
---------------------------------------->
Log session started: 04-10-14, 08:27:19 AM
Platform: WinNT 4.0
Release Compilation
pszPrinterName ImageMaker DocCnvrt Driver
pszAnsiFileName C:\CommServer\unityMta\IpFaxScratchpad\MsgBody.txt
pszOutFileName C:\CommServer\unityMta\IpFaxScratchpad\render.tif
FileType:0 multiPages:1 command:2 timeout:1800

Entering pdcf_WriteInfoForControlDialog()
oppening file: "C:\ImgMaker\Conversion_Toolkit\ImageMaker DocCnvrt Driver.FWD"
to transfer data (printdoc.dll -> printdoc.exe dialog)
data flushed!
MS: 18808737
```

```
deleted file: "C:\ImgMaker\Conversion_Toolkit\ImageMaker DocCnvrt Driver.BWD"
used to transfer data from dialog back to here!
MS: 18808737

Leaving pdcf_WriteInfoForControlDialog()

verb:[printto] params:["ImageMaker DocCnvrt Driver" "winspool" "Ne00:"]
lpFile:[C:\CommServer\unityMta\IpFaxScratchpad\MsgBody.txt]

Calling ShellExecuteEx()
MS: 18808737
ShellExecute() succeded

Calling pdcf_WaitForDoneFromControlDialog()
MS: 18808769

Entering pdcf_WaitForDoneFromControlDialog()
Monitoring cancel event: ImageMaker DocCnvrt Driver.printdoc.cancel
Event already exists [this is good]
  fopen() failed
  fread() failed
!DONE: timeout = 1799
reading file: "C:\ImgMaker\Conversion_Toolkit\ImageMaker DocCnvrt Driver.BWD"
to get 'done' signal from dialog
MS: 18809776
ppd_trans_feedback.done = TRUE

Calling pdcf_WriteAckToControlDialog()
MS: 18809776

Entering pdcf_WriteAckToControlDialog
flushing 'done ack' signal to dialog
MS: 18809776
Have to wait

Leaving pdcf_WriteAckToControlDialog


Leaving pdcf_WaitForDoneFromControlDialog()

pdcf_WaitForDoneFromControlDialog() returned
MS: 18810425

Entering pdcf_CleanUpControlDialog()
MS: 18810425
Log session stopped: 04-10-14, 08:27:21 AM
------------------------------------->

------------------------------------->
Log session started: 04-10-14, 08:27:21 AM
Platform: WinNT 4.0
Release Compilation
pszPrinterName ImageMaker DocCnvrt Driver
pszAnsiFileName C:\CommServer\unityMta\IpFaxScratchpad\products.doc
pszOutFileName C:\CommServer\unityMta\IpFaxScratchpad\render.tif
FileType:0 multiPages:1 command:2 timeout:1800

Entering pdcf_WriteInfoForControlDialog()
oppening file: "C:\ImgMaker\Conversion_Toolkit\ImageMaker DocCnvrt Driver.FWD"
to transfer data (printdoc.dll -> printdoc.exe dialog)
data flushed!
MS: 18810518
deleted file: "C:\ImgMaker\Conversion_Toolkit\ImageMaker DocCnvrt Driver.BWD"
used to transfer data from dialog back to here!
MS: 18810518
```

```
Leaving pdcf_WriteInfoForControlDialog()

verb:[printto] params:["ImageMaker DocCnvrt Driver" "winspool" "Ne00:"]
lpFile:[C:\CommServer\unityMta\IpFaxScratchpad\products.doc]

Calling ShellExecuteEx()
MS: 18810518
ShellExecute() succeded

Calling pdcf_WaitForDoneFromControlDialog()
MS: 18810561

Entering pdcf_WaitForDoneFromControlDialog()
Monitoring cancel event: ImageMaker DocCnvrt Driver.printdoc.cancel
Event already exists [this is good]
  fopen() failed
  fread() failed
!DONE: timeout = 1799
reading file: "C:\ImgMaker\Conversion_Toolkit\ImageMaker DocCnvrt Driver.BWD"
to get 'done' signal from dialog
MS: 18811581
!DONE: timeout = 1798
reading file: "C:\ImgMaker\Conversion_Toolkit\ImageMaker DocCnvrt Driver.BWD"
to get 'done' signal from dialog
MS: 18812585
ppd_trans_feedback.done = TRUE

Calling pdcf_WriteAckToControlDialog()
MS: 18812585

Entering pdcf_WriteAckToControlDialog
flushing 'done ack' signal to dialog
MS: 18812585
Have to wait

Leaving pdcf_WriteAckToControlDialog


Leaving pdcf_WaitForDoneFromControlDialog()

pdcf_WaitForDoneFromControlDialog() returned
MS: 18812858

Entering pdcf_CleanUpControlDialog()
MS: 18812858
Log session stopped: 04-10-14, 08:27:23 AM
-------------------------------------->

-------------------------------------->
Log session started: 04-10-14, 08:27:23 AM
Platform: WinNT 4.0
Release Compilation
pszPrinterName ImageMaker DocCnvrt Driver
pszAnsiFileName C:\CommServer\unityMta\IpFaxScratchpad\cover.htm
pszOutFileName C:\CommServer\unityMta\IpFaxScratchpad\
0188c93ff8194177b9f313edabc2ea73.tif
FileType:0 multiPages:1 command:2 timeout:1800

Entering pdcf_WriteInfoForControlDialog()
oppening file: "C:\ImgMaker\Conversion_Toolkit\ImageMaker DocCnvrt Driver.FWD"
to transfer data (printdoc.dll -> printdoc.exe dialog)
data flushed!
MS: 18812890
deleted file: "C:\ImgMaker\Conversion_Toolkit\ImageMaker DocCnvrt Driver.BWD"
used to transfer data from dialog back to here!
```

```
MS: 18812890

Leaving pdcf_WriteInfoForControlDialog()

verb:[printto] params:["ImageMaker DocCnvrt Driver" "winspool" "Ne00:"]
lpFile:[C:\CommServer\unityMta\IpFaxScratchpad\cover.htm]

Calling ShellExecuteEx()
MS: 18812894
ShellExecute() succeeded

Calling pdcf_WaitForDoneFromControlDialog()
MS: 18812897

Entering pdcf_WaitForDoneFromControlDialog()
Monitoring cancel event: ImageMaker DocCnvrt Driver.printdoc.cancel
Event already exists [this is good]
  fopen() failed
  fread() failed
!DONE: timeout = 1799
  fopen() failed
  fread() failed
!DONE: timeout = 1798
reading file: "C:\ImgMaker\Conversion_Toolkit\ImageMaker DocCnvrt Driver.BWD"
to get 'done' signal from dialog
MS: 18814909
ppd_trans_feedback.done = TRUE

Calling pdcf_WriteAckToControlDialog()
MS: 18814909

Entering pdcf_WriteAckToControlDialog
flushing 'done ack' signal to dialog
MS: 18814909
Have to wait

Leaving pdcf_WriteAckToControlDialog


Leaving pdcf_WaitForDoneFromControlDialog()

pdcf_WaitForDoneFromControlDialog() returned
MS: 18815620

Entering pdcf_CleanUpControlDialog()
MS: 18815620
Log session stopped: 04-10-14, 08:27:26 AM
-------------------------------------->

-------------------------------------->
Log session started: 04-10-14, 08:27:26 AM
Platform: WinNT 4.0
Release Compilation
pszPrinterName ImageMaker DocCnvrt Driver
pszAnsiFileName C:\CommServer\unityMta\IpFaxScratchpad\render.tif
pszOutFileName C:\CommServer\unityMta\IpFaxScratchpad\
0188c93ff8194177b9f313edabc2ea73.tif
FileType:0 multiPages:1 command:2 timeout:1800

Entering pdcf_WriteInfoForControlDialog()
oppening file: "C:\ImgMaker\Conversion_Toolkit\ImageMaker DocCnvrt Driver.FWD"
to transfer data (printdoc.dll -> printdoc.exe dialog)
data flushed!
MS: 18815632
deleted file: "C:\ImgMaker\Conversion_Toolkit\ImageMaker DocCnvrt Driver.BWD"
used to transfer data from dialog back to here!
```

```
     MS: 18815632

     Leaving pdcf_WriteInfoForControlDialog()

     verb:[printto] params:["ImageMaker DocCnvrt Driver" "winspool" "Ne00:"]
     lpFile:[C:\CommServer\unityMta\IpFaxScratchpad\render.tif]

     Calling ShellExecuteEx()
     MS: 18815632
     ShellExecute() succeded

     Calling pdcf_WaitForDoneFromControlDialog()
     MS: 18815659

     Entering pdcf_WaitForDoneFromControlDialog()
     Monitoring cancel event: ImageMaker DocCnvrt Driver.printdoc.cancel
     Event already exists [this is good]
       fopen() failed
       fread() failed
     !DONE: timeout = 1799
     reading file: "C:\ImgMaker\Conversion_Toolkit\ImageMaker DocCnvrt Driver.BWD"
      to get 'done' signal from dialog
     MS: 18816667
     !DONE: timeout = 1798
     reading file: "C:\ImgMaker\Conversion_Toolkit\ImageMaker DocCnvrt Driver.BWD"
      to get 'done' signal from dialog
     MS: 18817671
     ppd_trans_feedback.done = TRUE

     Calling pdcf_WriteAckToControlDialog()
     MS: 18817671

     Entering pdcf_WriteAckToControlDialog
     flushing 'done ack' signal to dialog
     MS: 18817671
     Have to wait

     Leaving pdcf_WriteAckToControlDialog


     Leaving pdcf_WaitForDoneFromControlDialog()

     pdcf_WaitForDoneFromControlDialog() returned
     MS: 18818319

     Entering pdcf_CleanUpControlDialog()
     MS: 18818319
     Log session stopped: 04-10-14, 08:27:29 AM
     ------------------------------------->
```

Q.931 Disconnect Cause Codes

```
     Code        Cause
        0           Valid cause code not yet received
        1           Unallocated (unassigned) number
        2           No route to specified transit network (WAN)
        3           No route to destination
        4           send special information tone
        5           misdialled trunk prefix.
        6           Channel unacceptable
        7           Call awarded and being delivered in an established channel
        8           Prefix 0 dialed but not allowed
        9           Prefix 1 dialed but not allowed
        10          Prefix 1 dialed but not required
        11          More digits received than allowed, call is proceeding
```

```
16          Normal call clearing
17          User busy
18          No user responding
19          no answer from user
21          Call rejected
22          Number changed
23          Reverse charging rejected
24          Call suspended
25          Call resumed
26          Non-selected user clearing
27          Destination out of order
28          Invalid number format (incomplete number)
29          Facility rejected
30          Response to STATUS ENQUIRY
31          Normal, unspecified
33          Circuit out of order
34          No circuit/channel available
35          Destination unattainable
37          Degraded service
38          Network (WAN) out of order
39          Transit delay range cannot be achieved
40          Throughput range cannot be achieved
41          Temporary failure
42          Switching equipment congestion
43          Access information discarded
44          Requested circuit channel not available
45          Pre-empted
46          Precedence call blocked
47          Resource unavailable – unspecified
49          Quality of service unavailable
50          Requested facility not subscribed
51          Reverse charging not allowed
52          Outgoing calls barred
53          Outgoing calls barred within CUG
54          Incoming calls barred
55          Incoming calls barred within CUG
56          Call waiting not subscribed
57          Bearer capability not authorized
58          Bearer capability not presently available
63          Service or option not available, unspecified
65          Bearer service not implemented
66          Channel type not implemented
67          Transit network selection not implemented
68          Message not implemented
69          Requested facility not implemented
70          Only restricted digital information bearer capability
                   is available
79          Service or option not implemented, unspecified
81          Invalid call reference value
82          Identified channel does not exist
83          A suspended call exists, but this call identity does not
84          Call identity in use
85          No call suspended
86          Call having the requested call identity has been cleared
87          Called user not member of CUG
88          Incompatible destination
89          Non-existent abbreviated address entry
90          Destination address missing, and direct call not subscribed
91          Invalid transit network selection (national use)
92          Invalid facility parameter
93          Mandatory information element is missing
95          Invalid message, unspecified
96          Mandatory information element is missing
97          Message type non-existent or not implemented
98          Message not compatible with call state or message type
                non-existent or not implemented
```

```
99          information element nonexistant or not implemented
100         Invalid information element contents
101         Message not compatible with call state
102         Recovery on timer expiry
103         parameter non-existent or not implemented - passed on
111         Protocol error unspecified
127         Internetworking, unspecified
```

Related Information

- **Voice Technology Support**
- **Voice and Unified Communications Product Support**
- **Troubleshooting Cisco IP Telephony** ⬏
- **Technical Support & Documentation – Cisco Systems**

Updated: Jul 02, 2007   Document ID: 63355