

# **Rossware User's Email Integration Handbook**

First, a distinction: When using the word “email,” this document is not referring to ServiceDesk’s *internal* mail system. We know users sometimes employ that word when referring to those within-ServiceDesk communiqués. For clarity, at Rossware we try to refer to those as “SD-Mails” – reserving the word “email” for the animals you send across the internet to (or receive from) other people.

Several of Rossware’s applications use email, whether it’s for sending purposes (typically the case), or receiving (as with the *Emailed Dispatch Receiver*). This document is provided to help you assure your system is setup to successfully integrate with Rossware emailing functions.

## **Chapter 1**

---

### **General Concepts**

Historically, Rossware systems did not do any emailing directly. Instead, they requested that any email-related service be provided by Windows. In 2010 and 2011, we added a new option whereby many Rossware systems can perform email tasks directly. Regardless, we have simultaneously retained the old mechanisms, which means you have a choice: stay with the old/default systems that employ Windows, or opt to use the new/direct methods instead.

So that you can more fully understand the difference between, suppose you are using the old method. Suppose further you have an email program (e.g., Outlook, Outlook Express, Windows Live Mail, etc.) appropriately setup on your computer. With this as foundation, you double-click on an email address in ServiceDesk to request initiation of an email to that addressee as the recipient. Shortly, you will see a compose-email window, where you can type in your message and then send. Do you suppose ServiceDesk created this compose-email window for you? It did not. In fact, the sole extent of its involvement was, when you did that little double-click, it made a “call” to Windows. Essentially, it “called out” as follows:

“Hey, Windows, please open a compose-mail window with [x@xx.net](mailto:x@xx.net) as the addressee.” [using, of course, the actual email address in question]

That’s all ServiceDesk did. The rest was done downstream, based on a takeover of the request by Windows.

This raises an important question. What does Windows do, when it receives a request like this.

The answer is simple. Windows says to itself:

"Hmmm, this looks like a request that needs to be fulfilled by an *email program*. I guess I should relay it onward to an email program. But, which program? Ahhh, I know: the user has designated xxxxxxxx [where xxxxxxxx is the actual program for which this is true] as his/her *default* program, so I'll pass the request onward to it."

And that's precisely what Windows does. It relays the request to whatever email program is designated, in Windows, as its *default* email program (sometimes called the "email client," or "Windows Mail Client"). It's then up to that program to do the actual work.

Thus (and for example), if you have Outlook Express designated as your default email program, it will actually be Outlook Express that presents your compose-email window (and which, as well, will manage your interaction from that point forward). The sole involvement of ServiceDesk was to make the initiating request to Windows.

Our new direct methods, by contrast, manage email processes via their own, internal mechanisms. Thus, they involve no call to Windows, and make no use of any underlying mail client. They were created to bypass a number of challenges that have sometimes cropped up when using Windows and its mail client. It's not that these challenges have not been ultimately surmountable; it's just that the new, Rossware-Direct methods can often be a much more straight-forward path to success.

## ***Chapter 2***

---

### **Using the Windows Mail Client**

As of May 2011, we have enabled Rossware direct-mail methods in all Rossware systems except the tech-side of SD-Mobile. In general, we think you're going to be much happier using the direct methods, and encourage you to go in that direction. Even so, you have a choice, and if you're determined to integrate with the Windows Mail Client, this chapter is for you (otherwise, please feel perfectly justified in skipping over it entirely).

#### **A. Setup of Your Window Mail Client**

When a Rossware application makes its "call" to Windows requesting a particular email service, it uses a language, in Windows, called MAPI (mail applications programming interface). For Windows to be able to properly relay the request to an email program, it needs a program that is able to speak the same language. Thus, it's important that you have an email program installed that is MAPI-compliant. Most are, including Outlook Express (often called OE), Outlook, Windows Live Mail and Thunderbird.

Of course, we're talking *email programs* here, and if you're a neophyte a little explanation might help.

If you've been managing your emails on-line via a service such as Hotmail, Gmail, Yahoo or similar, please understand those are not email programs. Rather, they are

on-line, browser-based interfaces for managing your email — *animals of a very different sort.*

With an installed *program*, by contrast, you’re working with an application that’s installed right on your local machine. This means your direct interaction is all local, and for that reason does not depend on a *live and simultaneous* internet connection. Ultimately you do need a connection, because one the program’s tasks (aside from interacting with you) is to occasionally connect to the internet, and download emails as sent to you, plus upload any you are sending out (this is in contrast to an on-line interface, where the interaction is all managed by some other entity’s server somewhere). It *does* require a particular kind of email account (typically called POP3) — one that’s designed to accommodate such direct downloading and uploading.

Hopefully, you get the difference. It’s about a locally-installed and running program, versus an on-line window of management. That latter will not integrate with Rossware applications. It *requires* the former.

If you’ve formerly been limited to use solely of a web-based mail management system, don’t worry about any cost as connected with setting up an installed system.

First, you’ll need an email account that’s appropriate to the purpose, and . . . guess what? You’re already paying for some quantity of such accounts. Virtually all internet service plans include several “free” such accounts (which, perhaps, you’ve simply never used). It’s time to change that. Call your ISP (that’s geek-speak for internet service provider: it’s the company you buy your internet access from) about activating one or more of your “free” accounts. While you’re at it, ask for assistance in setting up your email program to work with an account that’s being activated.

As for that actual setup, it’s a simple matter of putting into your email program the particular values that pertain to the account it will be connecting to. You’re telling it, in other words, how to talk to your ISP-provided email account. If you know the right values and where to put them, it’s very simple. For anyone that’s remotely geekish, this is very easy. Otherwise, a walk-through by a help-person at your ISP will make it easy for you, too.

As for the installed program of choice, again, it must be MAPI-compliant. Beyond that, for an email program that’s going to integrate operations with other applications, we generally prefer the Windows-included options as being the most issue free (Outlook Express was included with all Windows versions up to XP, versions since have included Windows Live Mail).

## **B. Assuring Your Email Program is Set as the Windows Default**

As mentioned, when a Rossware application makes its request to Windows, Windows’ task is to then relay that request to an actual program. Windows choice as to the target of this relay process will be to consult, within its own settings, to see which email program is designated as “default.” Thus, to assure integration from software is with the

email program you intend, you must assure it's set as the Windows default. Here are the appropriate steps:

In Windows XP, it's done from Internet Explorer. Open that program, click on '*Tools*,' then '*Internet Options*,' then the '*Programs*' tab. In the '*E-mail*' box, select '*Outlook Express*,' and click '*OK*.'

In Windows Vista and Windows 7, go to the '*Control Panel*', then '*Default Programs*'. Select the email program you want, then enable each of the applicable email functions.

Sometimes, we've discovered, Windows needs to be rebooted before a default selection truly takes operative effect.

### **C. Coping with Email Security**

Computer viruses seek to replicate. One of their more common strategies is to email themselves to other computers, using your installed email program. As one measure against this, most email programs distinguish between an email request as initiated by a live user, versus one as initiated by a program that's running in the system. They treat the latter as being much more suspect (especially if the request is to send a mail without intervening user review). More specifically, when they detect a program making such a request, they will generally display a Window asking for live-user approval, before proceeding.

The above strategy is great, assuming you don't have programs that need to have automated access to your email, especially for un-reviewed sending purposes. If you're extensively using Rossware systems, however, that is decidedly not the case. We have a plethora of functions that ease human burdens by initiating email access for you, programmatically — and you certainly don't want to be pestered with a request for permission each time one of these actions is performed.

For that reason (and at least if you're using any Rossware systems that do direct email sends), this element of security positively must be circumvented (if you're not doing direct email sends; you'll likely be fine regardless).<sup>1</sup>

In Outlook Express or Windows Live Mail, this is easy. Within the email program, go to *Tools*, then *Options*, then look for the checkbox that reads: "*Warn me when a program tries to send mail as me*" (potentially, the wording may vary, but should mean essentially the same thing). Uncheck that box, and save your settings. That's it. With so simple a change, you should now have great email integration with Rossware applications.

If, on the other hand, you have chosen to make Microsoft's Outlook your integrated email program, you'll not find coping with security is so simple. The simple problem is

---

<sup>1</sup> Examples of direct email sends are: (1) ServiceDesk emailing appointment confirmation requests; (2) ServiceDesk sending report that parts have arrived, and requesting the customer link or call to schedule the completion visit (this context can be with intervening user review or without, depending on user choice); (3) SD-MobileLink emailing ticket images to the customer; (4) SD-CyberLink emailing confirmation of scheduled service.

that, unlike the other programs, Outlook does not offer the option of turning off the warning. The option simply is not there.

If you feel you must use Outlook as your preferred email program, there were, historically, two possible solutions:

1. Maintain use of Outlook as the email program in which you manage your direct/active email tasks (opened via a direct shortcut for the purpose), but make one of the others the Windows default program. Remember, it's the default program Rossware apps will integrate to, so you can essentially be using two email programs at once.
2. Use *ClickYesPro* or similar. This is, simply, a program that runs constantly in the background and pretends to be a live user, consenting to the email program's request for user consent when the request is presented. Some users have had good success with it. Some have not. If you want to try it, just Google ClickYesPro.

Again, those were the *historical* options. As of November 2010, our Rossware-Direct method is likely to be much more favored.

Regardless, we must emphasize: If you're employing any of the processes that do direct email sends (i.e., sends that do not involve intervening user review), it's imperative to assure implementation of some solution — some method to prevent Rossware direct email attempts from being interdicted by requests for user consent. It simply is not practical to have a user constantly on-station to provide that consent, each and every time it's requested.

#### **D. Troubleshooting Integration with a Windows Mail Client**

Inevitably, when there are issues involving Rossware systems integration with email, it's because some element as above-described, within this document, has not been setup as prescribed.

Please bear in mind that (aside from the Rossware-Direct-Send method described in the next chapter) Rossware systems are not doing the actual email functions. They are making those very simple calls to Windows. If Windows returns an error in response to such a call (in which case the Rossware system will then relay that error to you), it is all but certainly because Windows could not fulfill the request — because some element of the setup, again, is not as above-prescribed.

It is your responsibility as a user to setup your email system — in essence, to give the Rossware boat water to float in.

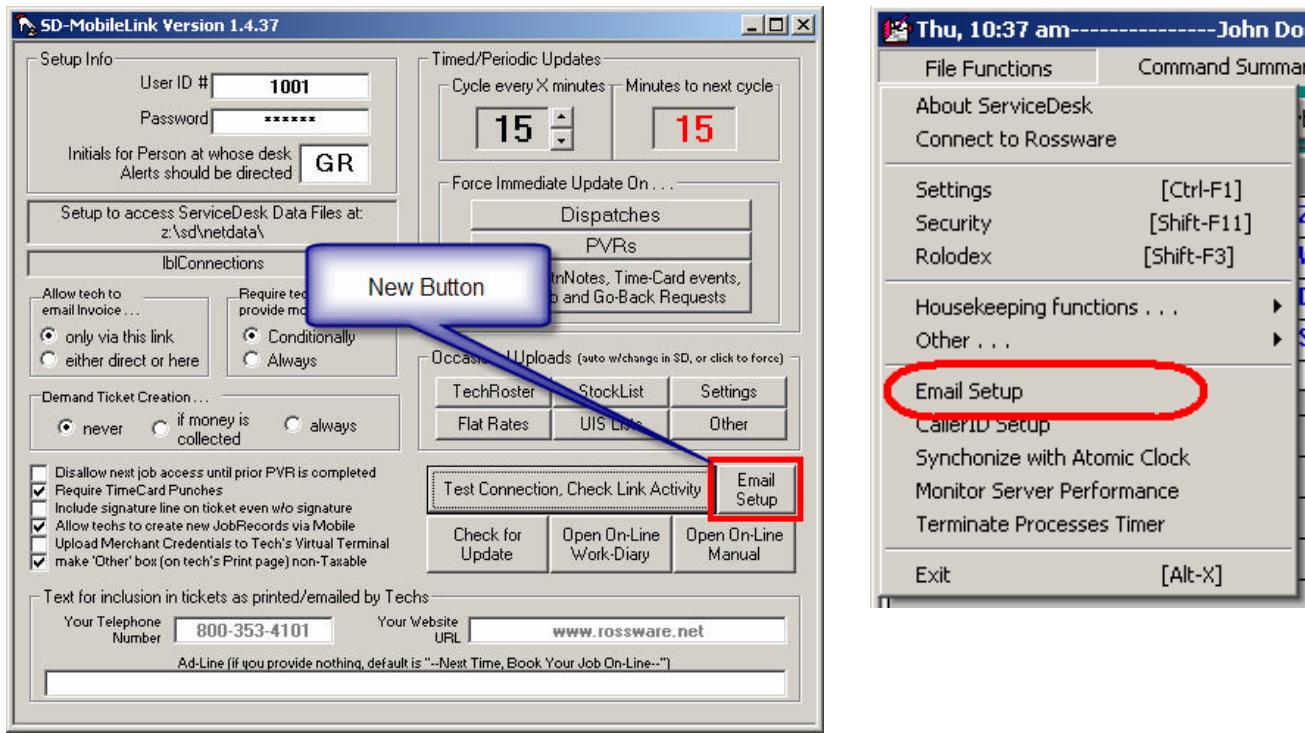
# Chapter 3

## Using the Rossware-Direct Method

As mentioned, the Rossware-Direct email option was created to provide an easier and more direct method by which to cope with issues as encountered when using a Windows mail client. As of May 2011, it's available as an option for all in-office Rossware applications that use email, except the tech-side of SD-Mobile.

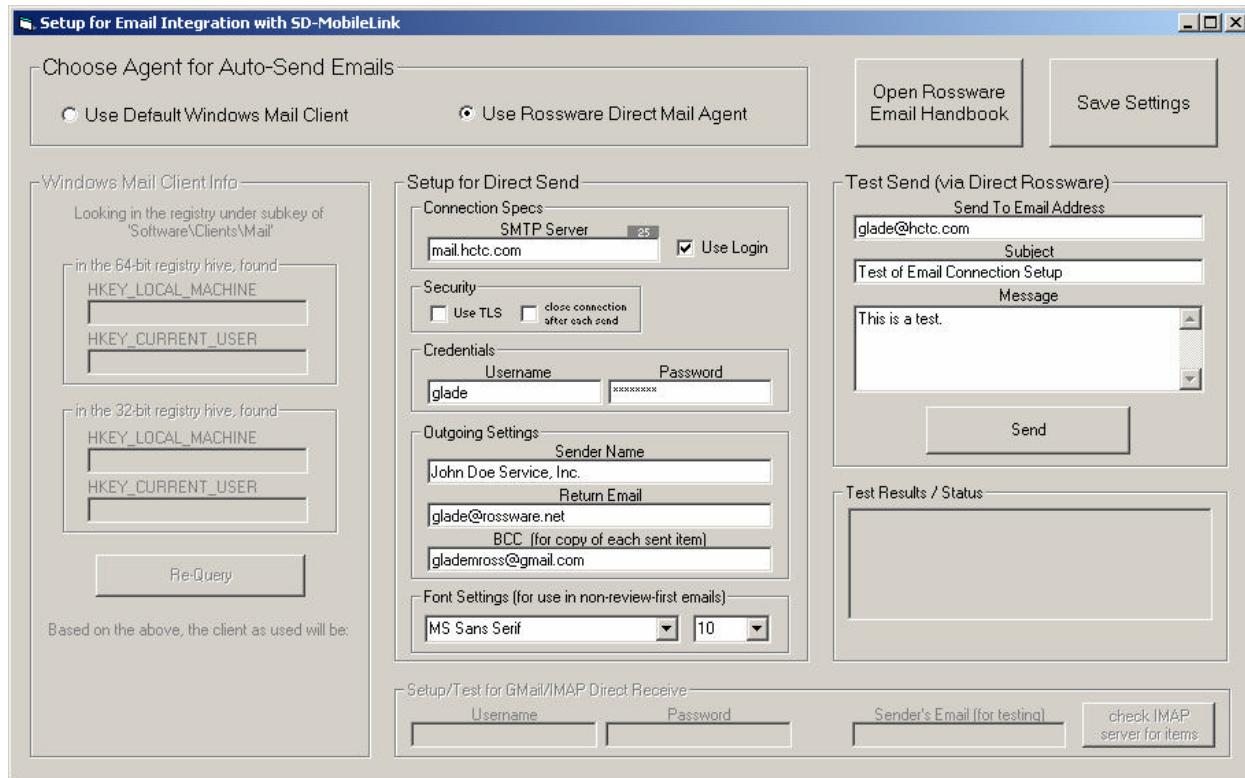
### A. Basic Setup

Each application will have a button (or similar element) via which to bring up the Email setup form. Below on the left, we show you where the button is located In the *SD-MobileLink* program. On the right, we show you where it's located in ServiceDesk itself:



The *CyberLink* program has a button similar to MobileLink's, as shown above-left, as does the *EmailedDispatchReceiver*. You should also be aware that from any such context, the setup is exclusive to that context. In other words, if you set to use the Rossware-Direct method in ServiceDesk, it will not change processes in MobileLink or CyberLink. Each must be set on its own.

In such regard, when you click on the applicable button from any such context, it will reward you with the following interface:



As you can see, this form offers two major options in the section at top: to have the underlying application use the Windows Mail Client (which is the default) versus using the new Rossware-Direct Mail Agent (which is what's actually set in the image as shown).

If you're using the default, there is a section (leftward in the form) that shows what is found in your Windows registry in regard to what is there-indicated as the default email program (and which, therefore, you can expect Windows to use as its mail client).

If instead you click to use the new Direct option, another section will enable (middle of form, and as above shown) where you may fill-in the needed specs via which the system will connect to your email account (the specs as above shown are just for illustration purposes, your needed specs will be particular to your own specific email account).

In either case, the third (right-most) section allows you to do test sends (it will do the test based on method and parameters you've specified to the left).

One detail that may not be obvious, in terms of intended use, is the “BCC” box (lower center). If you're not aware, BCC stands for “blind carbon copy,” and in *ordinary* email contexts the notion behind a BCC is you can “copy” one or more other parties without your main recipient (or even standard “CC” recipients) being aware you're doing so. In this context, though we have another purpose for the BCC. It is to compensate for the following:

When using an ordinary email program, your sent items are typically copied for you into a “Sent Items” folder. Based on this, if you ever want to see what’s been sent, all you must do is look in that folder. Our Rossware-Direct mail agent has no such folder. Our thinking is you can use the BCC box to accomplish a near equivalent. Specifically, if you place an email address there, any item that the Direct system sends out will also be sent to that BCC address. Make that BCC address one of your own and you will thus have a copy of what was sent. You may, in fact, configure your standard email program (i.e., the one that handles your incoming emails) to automatically detect items as are direct-sent in this context, and to automatically place such emails in a particular folder within it, as setup for the purpose.

Aside from the above little complication, we think overall it’s quite self-explanatory — at least if you’re even a little bit savvy about email setups. If you’re not, we’ll provide a tiny explanation here.

While the Rossware-Direct Mail Agent provides a Windows-Client-Free method of accessing email, it does not provide an actual email account. You still must have such an account (provided by your internet provider, Gmail, Hotmail, Yahoo or similar) for it to connect to, and the purpose of those boxes (that you must fill-in, if wanting to use the direct agent) is to inform it of what are applicable details on that account. It’s sort of like, if you want it to call a particular phone, it needs the phone number.

The most pertinent such detail, that must be filled-in, is the “SMTP Server” designation. This is, simply, the *internet call-name* for the “server” that accepts emails as being sent via your applicable email account. If you’re using Gmail, for example, the correct SMTP Server is “*smtp.googlemail.com*”. If you’re using Comcast, it’s “*mail.comcast.net*”. You simply need to determine, from whomever provides the email account you’re using, what is the proper SMTP Server for your account, and stick it in the indicated box.

Of course, you also need to put in the proper username and password as applies on that account. Another tiny detail is the “Use TLS” checkbox. Most servers should not have this box checked (Gmail is an exception). Regardless (and assuming you don’t know), it’s easy to test with it one way, and if you don’t have good results, test the alternate.

Just understand that whatever you set and save here is precisely what the applicable Rossware program will use, when handling emails for you. If you can do a successful test send with whatever you’ve set (and, also, assuming you save that setup), it’s pretty much a guarantee the applicable Rossware program will have no issues when doing real work.

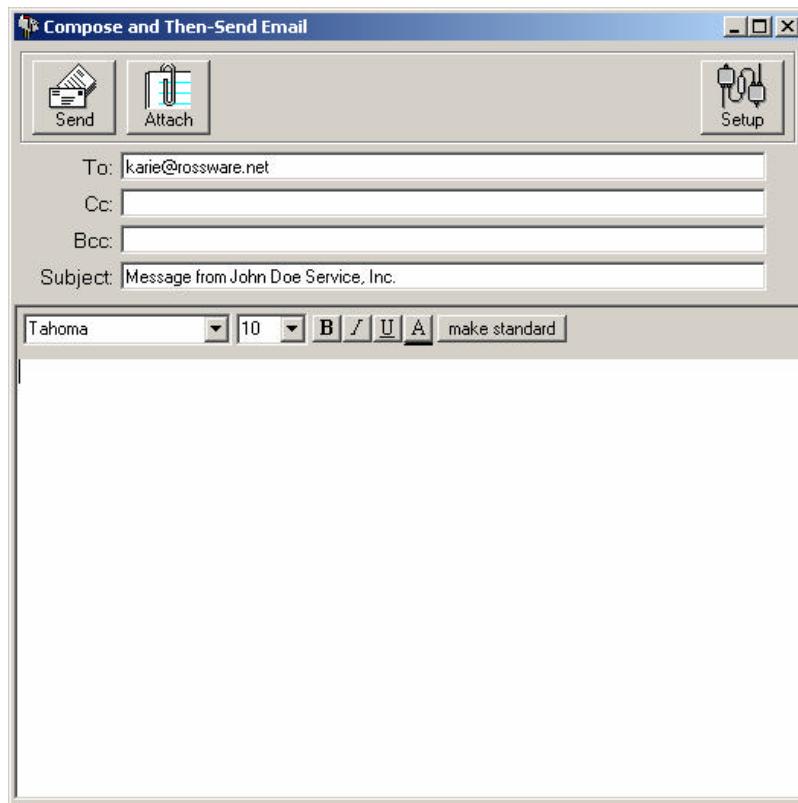
In fact, what follows here is all extra. Basic direct-email functionality will be assured based on you doing nothing more than the minimal setup, as above-described. You should bother reading further here only if you want to get more sophisticated, and with deeper understanding.

## **B. Using the Integrated Compose-Mail Window**

When we initially introduced the Rossware direct-mail option in 2010, its use was limited to those contexts where emails are sent for you *without* any prior user review and/or editing (e.g., you request the “send,” and it simply goes, behind the scenes). The reason for the limitation was simple: while we’d programmed mechanisms to manage direct (non-Windows-mail-client-managed) sending, we’d not programmed anything to take the place of a Windows Mail Client’s compose window. I’m referring to the interface that opens when you want to type up a new email. We’d made no interface like that. Absent having such an interface, the only way to give you that experience, in contexts where you want it, was to continue calling on the Windows mail client. This was the state of programming until April 2011, and release of ServiceDesk Ver. 4.5.13.

With that release and forward, however, and if you’ve setup within your *Email Setup Options* form to use the direct method, ServiceDesk will use the direct method consistently and in all sending contexts, including “review-first” emails. This is accomplished by virtue of the fact we have now created our own “compose-email” interface. Again, assuming you’ve opted for the direct method, this will appear any time you initiate an email in a “review-first” context.

By way of example (and for easy testing), just type an email address into the email address box of a Callsheet, then double-click on it. If you’re setup to use the Windows Mail Client, you’ll see its compose window open. But if you’ve setup for the Rossware Direct method, you’ll see its own compose window open instead:

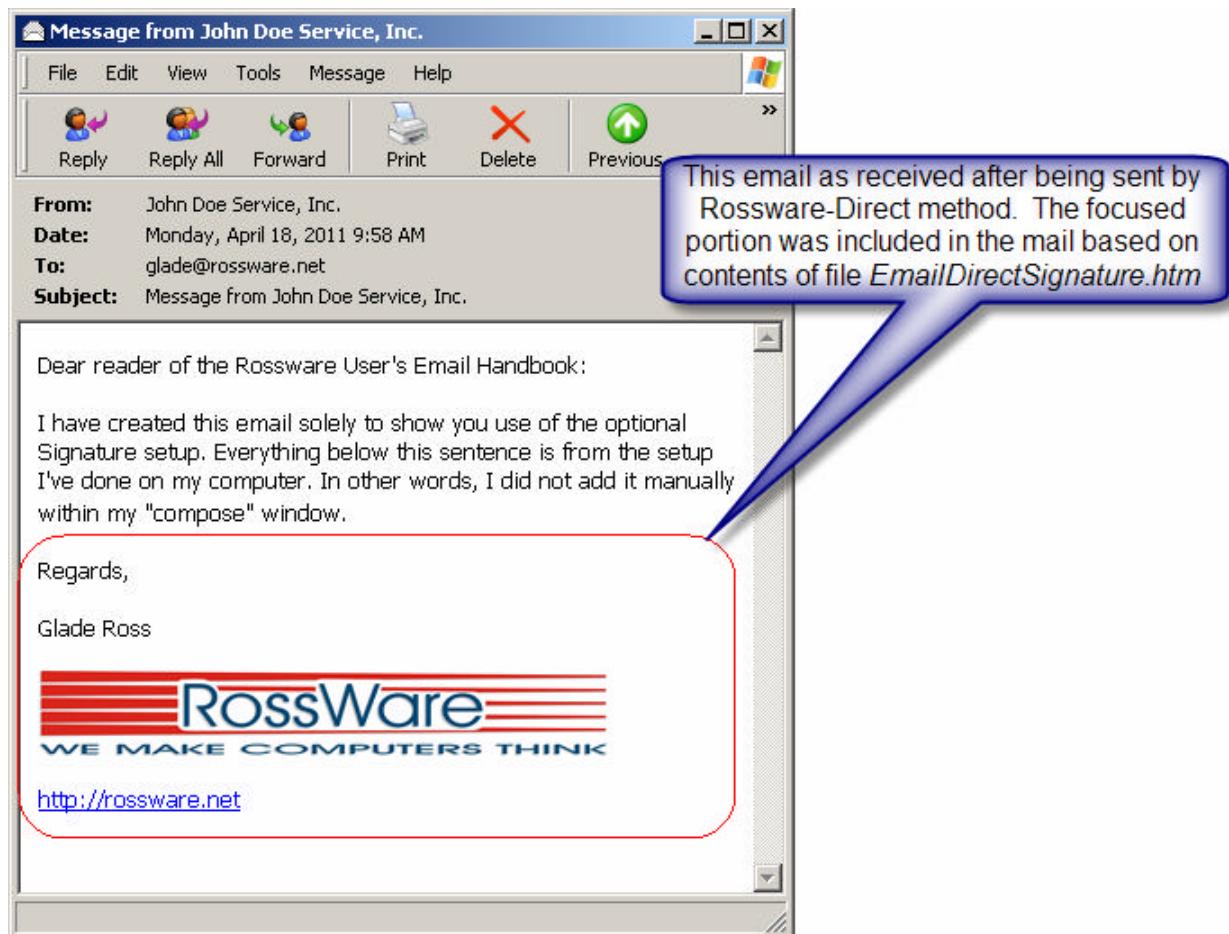


You should notice the appearance of this interface is very similar to what you are likely used to, if you've otherwise been using most any of the common Windows mail clients. Its operation is also very similar — so similar, in fact, we urge you to not be confused into thinking you're looking at a Windows Mail Client. You are not. This is solely and totally an internal-to-ServiceDesk, and created-by-Rossware form. All its machinery is managed by us.

We also think you'll find its operation very straightforward. If you learned to manage creating and sending of emails in other contexts, you should have no difficulty here.

### **C. Adding Customized “Signatures”**

One of the things you can do in an ordinary Windows Mail Client is configure it to add a “signature” space at the bottom of your emails. The signature space can include closing text, a logo for your business, perhaps a hyperlink to your website and telephone number — that kind of thing. It occurred to us you might want to do the same in conjunction with Rossware-Direct sent emails, so we created accommodation for it.



In fact, you may configure a series of different “signatures” for use in varying contexts. For example, you can setup just one, generalized-company signature that's used for any situation where a more specific signature has not been setup. As for more specific

such signatures, you may configure so that each user has his or her own. You may likewise configure so that each application has one specific to its use (one, for example, as configured for emails as sent by SD-MobileLink, and a different signature for emails as sent by SD-CyberLink). You can even make signatures that are specific to particular processes within applications (one, for example, for use when SD-MobileLink emails the customer a ticket, and another for when it emails a link for JobStatus-checking). You can be as general or specific as you like.

### i. Basic “Signature” Setup

For any signature you want, your simple task is to setup a file that defines it. The contents within this file must be in the HTML format (stands for “hyper-text-markup-language”). This is the same language that’s used to define web pages.

If you know your way around html (many people do, for many people work on their own web pages) the task will be very simple. If you do not, we suggest using a trick.

One trick would be to ask someone, who knows html, to do it for you. Again, a *lot* of people know html.

If that’s not the best trick for you, there’s another one that makes it very easy. Just create the signature you want within Microsoft’s Word program (or anything similarly equivalent). In other words, open a document, type the text you want, set the desired font characteristics, add any graphics you want, etc.

After you’ve made a document that contains exactly the content you want in your signature, save it with the name and in the location as applicable to its purpose (as defined in the next section). However, you’ll need to save it in a special way. By default, Word would normally save your document in *Word* format (which gives the filename a .doc or .docx extension). That’s not what you want. Instead, pay attention in the “Save As” dialog box. Down toward the bottom there’s a dropdown labeled “Save as type.” You want to open that dropdown and pick (as the “type” to save) “*Web Page (htm., html)*.” When you pick that as the “type,” it saves your work in that magical “html” format (which is precisely what we said at the beginning is what you need as the end result).

See, it’s a good trick. Without knowing a thing about html, you can create an excellent html document. Cool, huh!

### ii. Signature Selection and FileSpecs

As mentioned, we’ve configured the system to allow you to setup, if desired, an array of different signatures, as applicable to different contexts. Again, each signature is a file with contents that must be in html format. Within the \sd folder on your server is a sub-folder called \EmailCustomElements. With one exception (noted below), each of your signature files must be located within this folder. In addition, the signature files that you setup must each be named precisely according to the following scheme:

*MySignature\_CompanyGeneral.htm*

(if you have no desire for more specific-purpose signatures, you may choose to setup just this one; in the absence of a more specific purpose signature being provided, this is the one the system will choose to use in all instances)

*MySignatureForMobile\_EmailsGeneral.htm*

(this is for the general purpose of emails as sent by SD-MobileLink; if present, it will supersede use of the above signature for emails as sent by this specific program)

*MySignatureForMobile\_SendTickets.htm*

(this is for the specific purpose of emails as sent by SD-MobileLink to provide customers with a copy of their tickets; if present, it will for that purpose supersede both items above)

*MySignatureForMobile\_SendStatusCheck.htm*

(this is for the specific purpose of emails as sent by SD-MobileLink to provide customers with a Status-Check Link; if present, it will for that purpose supersede the top two items)

Please note that as we expand signature-usage into other applications, the above list is expected to grow. We suggest you check back as often as you have a desire to see if a new specific-purpose setup has been added. Aside from those above-enumerated, there is one more signature file you may use. This one is different because it is not placed into that *\sd\EmailCustomElements* folder. Instead, it must be placed into the *LocalData* folder as applicable to the person to whom it will apply:

*EmailDirectSignature.htm*

(if you provide it, this will be used for emails as sent within ServiceDesk via the Email-Compose window; the logic is to allow a signature that's particular to the user; in its absence, whatever signature as is more general will be used; or none at all if none is provided)

If you are uncertain about what *LocalData* folder is applicable to a particular user, just consult the *About ServiceDesk* form, within ServiceDesk as configured for that user. Toward its lower left is a section that defines several folder paths, including the one of interest.

### **iii. How to Manage Graphic *Images* within your “Signature”**

Though creation of a signature file is generally simple, we'll admit it becomes a little more complex if you include images. This is because it's the nature of the html format to *reference* graphics, rather than *including* them within one and the same file as the html itself. In other words, in and of itself, an inserted image (such as the “Rossware” logo in my example) remains a separate file.

Because of this, the decision to include graphics means that, necessarily, you'll pay the price of dealing with more than one file, even if for a single signature (i.e., you'll have the html code itself, plus a file for each referenced image).

While there are a couple of ways this multi-file situation *could* be handled, for the Rossware-Direct solution we've chosen to implement but one.

First, the method we have *not* accommodated is where an html-referenced image is included with the email *as an attachment*. I personally receive quite a few emails where this method is used and — frankly — I hate it. Invariably, upon seeing the symbol that tells me my email has attachments, I wonder what's attached, look, and suffer frustration upon finding its nothing but an image in the signature. I did not want to provide you with a solution that, as an email recipient, I personally abhor.

Instead, we have setup for use of the second of the possible methods. If you want to include one or more images within your email signature, simply configure your HTML to reference an appropriate underlying image file at a location that will be available to your email recipient. Typically, this means having the file available, for such reference, on your website.<sup>2</sup>

In the case of the example signature that I show a couple of pages back, for example, I placed a copy of the applicable image file (RosswareLogo.jpg) within the public\_html folder on the Rossware website. I then configured the applicable section of code in my *EmailDirectSignature.htm* file to read (highlighted section pertinent) as follows:

```
</style>
</head>
<body lang=EN-US link=blue vlink=purple>
<div class=Section1>
<p class=MsoNormal><span style='font-size:10.0pt'>&ampnbsp</span></p>
<p class=MsoNormal><span style='font-size:10.0pt'>Regards,</span></p>
<p class=MsoNormal><span style='font-size:10.0pt'>&ampnbsp</span></p>
<p class=MsoNormal style='margin-bottom:12.0pt'><span style='font-size:10.0pt'>Glade
Ross</span></p>
<p class=MsoNormal><span style='font-size:10.0pt'></span></p>
<p class=MsoNormal><span style='font-size:10.0pt'><br>
<a href="http://rossware.net/">http://rossware.net</a></span></p>
</div>
</body>
</html>
```

Let me confess: I personally do not know html, and the above (in general) looks foreign to me. In fact, I used the “create-in-Word-then-save-as-html” trick (as recommended a couple of pages back) to create everything you see above, *except* what’s highlighted. As part of that effort (and within the context of Word), I simply composed the text I wanted, and stuck in the wanted image. After saving the result in html, I re-opened the resulting file in NotePad. This shows me the actual, underlying text (i.e., unlike Word, it

---

<sup>2</sup> What happens is, as your recipient’s email program decodes the html to display your message, it sees the reference to the image, then reaches to the referenced location to grab it, and displays it precisely as you formatted. As long as your recipient’s program has access via the referenced path, everything works perfectly.

does not re-interpret the html code as webpage-like composition). It was not difficult to find where the code initially referenced the file locally, to replace with a reference to the file on our website, and then simply save the result.<sup>3</sup>

Assuming you want to have images in your signature, a similar strategy should work great for you, too.

#### **iv. “Signature” View within the Compose-Mail Window**

In an ordinary Windows Mail Client/Program like Outlook or Outlook Express, when you’re in the Email-Compose window creating an email, you’ll see any “signature” you’ve setup automatically show at the bottom of your email.

We’d have liked to do the same for our Rossware-Direct Email-Compose window, but, for reasons having to do with the underlying programming and data structures, we found it problematic.

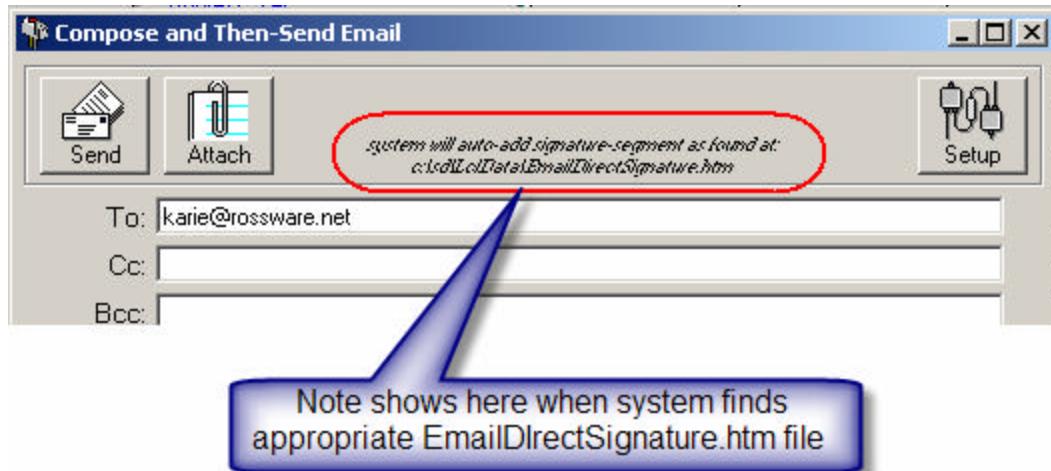
So, you don’t get that.

Instead, appending of the signature is done for you, as the email is sent (again, see my example for an illustration of potential result). What this means is, please don’t expect to see the signature within your actual compose window. Also, if you’ve configured your signature to include such closing niceties as “Regards,” it would be wise to avoid manually typing them within the body of your compose window; for otherwise your reader would end up seeing such niceties duplicated.

At the least, we do provide a contextual indicator, seeking to remind you if there’s no need to manually type in your own closing. If ServiceDesk finds an applicable *EmailDirectSignature.htm* file as applicable to the situation, it adds a little note section in the Compose form so indicating:

---

<sup>3</sup> Just in case it’s not clear, when either Word or a web browser opens an html file, it “interprets” the html as html, and displays it precisely as it’s meant to be seen. To put it another way, Word has an html-encoder built into it, and an html decoder too. NotePad, by contrast, has no html-decoder, so when it opens the file it shows you the plain-text un-interpreted code. It will save it in the same manner (i.e., as just the text that makes the code, which is precisely what you want).



The above might also be useful simply to verify, when you're seeking to place your file in proper position, if in fact you've succeeded.

#### **D. Use within the EmailedDispatchReceiver**

If you were perceptive, you might have noticed something about all the mail operations as discussed to this point: all involve *sending* emails; not *receiving*.

Of all Rossware applications, only one is involved in receiving emails: it's the *EmailedDispatchReceiver* (EDR), designed to automate reception of dispatches from entities that send them to you via email. It also was the last application we integrated with our direct-mail agent. Receiving emails is a very different process from sending them; we had to work out a whole other set of mechanisms for the purpose.

As with other Rossware applications, you continue to have the option to use the Windows Mail Client (see Chapter 2). However, with introduction within this application of the direct-agent option, we are discontinuing *support* for its use (specifically, within the EDR). This means, simply, if you want to use the old/Windows-Mail-Client method within the EDR, you're on your own. We'll not assist you in troubleshooting issues, there, unless you're using the new method.

We have deliberately designed the new method to be very trouble free, in part, by demanding you use a particular kind of email account for receiving the emailed dispatches it manages. Please note this is in contrast to the direct-agent's *sending* of emails function (i.e., in other applications), which allows you to use most any email account. For *receiving* dispatches, by contrast, we decided the benefit of no fuss performance outweighed giving you freedom on the matter. Thus, to use the direct-agent within the EDR, you must setup the particular kind of account we've programmed it to work with, and must instruct those entities who are emailing you dispatches (and whose dispatches you want grabbed by the EDR when working in direct-agent mode) to send your dispatches to that account.

What kind of email account must this be?

It must be a *Gmail* account, with *IMAP* enabled.

If you do not know, Gmail is Google's mail system, and you can setup and maintain the needed account for a very fair price. How does "FREE" grab you? Just go to <http://gmail.com>, follow the prompts, setup your account, then ask your particular clients who are applicable to start sending your dispatched emails to that account.

One slight complication involves enabling IMAP on that account. IMAP is a particular technology for email management and communication. It's the method our direct-mail-agent will use when grabbing emails from your Gmail account, so it's critical that you "turn on" the feature within your account. To do so, follow these steps:

1. Look for the tiny "gear" symbol in the top-right corner of your Gmail management webpage, and click on it.
2. In the dropdown, pick "*Mail Settings*."
3. In the large "Settings" page which then displays, look for the menu items that show along its top edge (they are similar to tabs). There you should see a "tab" labeled "*Forwarding and POP/IMAP*." Click on that tab.
4. Now you should see the actual page displayed, on which you can enable IMAP.

**Settings**

[General](#) [Labels](#) [Accounts and Import](#) [Filters](#) **Forwarding and POP/IMAP** [Chat](#) [Web Clips](#) [Labs](#) [Priority Inbox](#)

**Forwarding:** [Add a forwarding address](#)

Tip: You can also forward only some of your mail by creating a filter.

**POP Download:** [Learn more](#)

1. Status: **POP is enabled** for all mail that has arrived  
 Enable POP for **all mail** (even mail that's already been delivered)  
 Enable POP for **mail that arrives from now on**  
 **Disable** POP

2. When messages are accessed with POP: [keep Gmail's copy in the Inbox](#)

3. Configure your email client (e.g. Outlook, Eudora, Netscape Mail)  
[Configuration instructions](#)

**IMAP Access:** (access Gmail from other clients using IMAP) [Learn more](#)

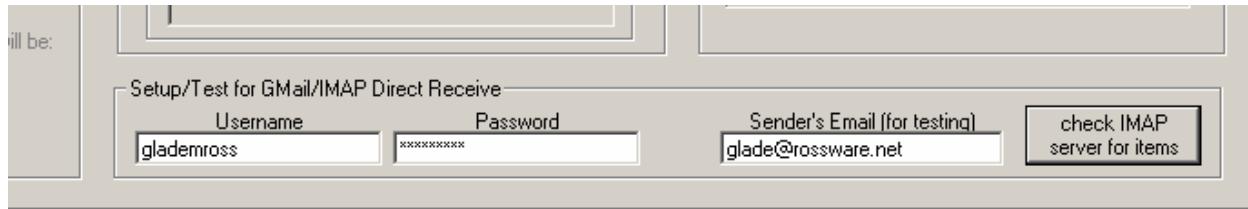
1. Status: **IMAP is enabled**  
 Enable IMAP  
 Disable IMAP

2. Configure your email client (e.g. Outlook, Thunderbird, iPhone)  
[Configuration instructions](#)

[Save Changes](#) [Cancel](#)

Once you've taken care of that setting and gotten applicable clients to send items to your new Gmail account, it's time to do the needed setup in your EDR program.

In such regard, you may have noticed, in the image of the Rossware Email-Setup window as shown on Page 7, there's a frame-area within the bottom-right that is (within the image as there shown) disabled. It's disabled in that view because that frame-area is solely concerned with setup as connected to the EDR's *receive* email function. For such reason, it's disabled when this window shows for any application *aside from* the EDR. If you open the setup window via the EDR, however, and also pick "Direct Mail Agent" at the top, this section will be enabled, and available for you to fill-in boxes, as follows:



As you can see, the fill-in is very simple (one of the advantages that accrues from our forcing you into using Gmail-solely for the purpose). All you must do is fill in your own Gmail username and password, then save. More than likely, though, you'll want test. For that purpose, send an email to your Gmail account (from *any* email place you can send), and in the above window's "Sender's Email (for testing)" box, place the return-email address for the account you sent the test email from. Make sure the test email remains in "Unread" status, and click on the "Check IMAP server for items" box. In the "Status" report box (part of that email-setup window), you should see a report indicating successful results (at least assuming you did the setup correctly).

It truly is easy, and avoids many of the complications that otherwise exist when using a Windows Mail Client. We highly suggest you go this route.

One further note, in regard to the EDR's method of grabbing emails: Whether you're using the Windows Mail Client or the Direct-Agent method, the EDR asks for emails that show as having been sent from the email address you specify, and specifically pays attention solely to those that are marked as "Unread." Please don't expect it to pay attention to any others. If there are emails you want it to grab, and they're already marked as having been read, you can mark them back to unread using mechanisms within the email interface.