



**IBM Rational ClearQuest EmailPlus Package
Administration Guide
Version 2.0**



Table of Contents

1 INTRODUCTION	4
2 PACKAGE DESIGN	4
2.1 RECORD TYPES	4
2.1.1 REPLICATED VERSUS NON-REPLICATED ENVIRONMENTS	5
2.1.2 CONFIGURING EMAIL ADDRESSES AND CLIENTS FOR CLEARQUEST USERS	6
2.2 EMAILPLUSCONFIG RECORD TYPE.....	6
2.3 EMAILPLUSSITECONFIG RECORD TYPE	7
2.3.1 EMAILPLUSSITECONFIG – SMTP TAB	7
2.3.2 EMAILPLUSSITECONFIG - CQ WEB TAB	11
2.4 EMAILPLUSRULE RECORD TYPE.....	12
2.5 EMAILPLUSTEMPLATE RECORD TYPE.....	12
2.5.1 EMAILPLUSTEMPLATE – TEMPLATE DEFINITION TAB.....	12
2.5.2 EMAILPLUSTEMPLATE – TEMPLATES TAB	14
2.5.3 EMAILPLUSTEMPLATE – USING RULES TAB	15
2.5.4 EMAILPLUSTEMPLATE – ATTACHMENTS TAB.....	16
2.6 EMAILPLUSACTION RECORD TYPE.....	16
2.7 EMAILPLUSRULESUBSCRIPTION RECORD TYPE	17
2.8 EMAILPLUSPOSTOFFICE RECORD TYPE.....	18
3 WORKING WITH EMAILPLUS TEMPLATES	18
3.1 TEMPLATE META TAGS	18
3.1.1 WORKING WITH EMBEDDED FIELD EXPANSIONS IN META TAGS	22
3.2 EMBEDDING IMAGES IN AN HTML BODY TEMPLATE	24
4 WORKING WITH EMAILPLUS RULES	24
4.1 INITIAL RULE DEFINITION	25
4.2 DEFINING A RULE PHRASE USING VALUE CRITERIA.....	25
4.3 DEFINING A RULE PHRASE USING FIELD CHANGES	29
4.4 DEFINING RULE PHRASES USING ACTIONS AND STATES.....	30
4.4.1 ACTION CONDITIONS.....	31
4.4.2 STATE TRANSITIONS.....	31
4.5 TO/CC ADDRESSES TAB.....	32
4.6 SUBSCRIPTION.....	33
4.7 DEFINING A RULE PHRASE USING AN ADVANCED RULE	35

5 THE ADMINISTRATOR GROUP	37
6 APPLYING THE EMAILPLUS PACKAGE	37
6.1 INSTALLING THE PACKAGE	37
6.2 APPLYING THE PACKAGE TO YOUR SCHEMA	39
6.3 ENABLING A STATELESS RECORD TYPE	40
6.4 CONFIGURING AN ENABLED RECORD TYPE	41
6.5 RECOMMENDED TESTING STRATEGY	42
6.5.1 COPY LIVE ENVIRONMENT	42
6.5.2 COPY SCHEMA.....	42
7 MAIL RELAY	43
7.1 SMTP RELAY	43
7.2 POSTOFFICE RELAY	44
7.3 EMP RELAY APPLICATION.....	44
8 DEPENDENCIES	47
9 PERFORMANCE	47
10 EMAIL ATTACHMENTS	48
10.1 DEFAULT CLEARQUEST CLIENT TEMPORARY CACHE DIRECTORY	48
10.2 SPECIFYING THE CLEARQUEST CLIENT TEMPORARY CACHE DIRECTORY	49
10.3 CLEARQUEST CLIENT TEMPORARY CACHE DIRECTORY HOUSEKEEPING	49
APPENDIX A – TEMPLATE META TAG SUBSTITUTION EXTENSION POINT	50
APPENDIX B – REGULAR EXPRESSIONS META CHARACTERS	52
APPENDIX C – FEATURES / MAIL TRANSPORT MATRIX	54
APPENDIX D – CONFIGURING ADDRESSING OPTIONS.....	56
APPENDIX E – EXAMPLE EMAILPLUS TEMPLATE EXPRESSIONS	58
APPENDIX F – EXAMPLE EMAILPLUSRULE ADVANCED RULES	60
APPENDIX G – DEBUGGING EMAILPLUS	61

1 Introduction

The ClearQuest EmailPlus Package (EmailPlus) exists to provide a more extensive solution to the problem of email notifications.

This guide describes version 2.0 (and possibly later versions) of the EmailPlus Package.

The functionality provided by the out-of-the-box ClearQuest Email Rules package, provides basic facilities to communicate record changes to users. EmailPlus provides a number of additional features to provide an enhanced email experience for users.

These features include support for:

- Customizing the content of the message subject and body texts to allow boilerplate text to be included.
- Including dynamic content in emails in addition to field old or new values.
- Including rich content (HTML).
- Providing access to SMTP restricted servers.
- Attaching files to emails.
- Allowing enhanced conditions to construct the rules.
- Allowing self subscription of users to rule or record notifications.
- Enabling enterprise-wide configuration of email notifications in addition to allowing individuals to turn on or off email notifications.
- Providing a number of performance improvements, particularly if queries are used.
- Extending the rules and data displayed through customer provided extensions.

2 Package Design

The EmailPlus package is implemented in PERL as a ClearQuest package. The package can be added to any ClearQuest schema. Although it is implemented entirely in PERL, it can be used to enable VBScript based schemas as packages have their own scripting language defined. There is also a restriction in VBScript based schemas with respect to stateless record types – See Section 6.3 for more details.

Care has been taken to ensure that the possibilities of name clashes are minimized by prefixing routine names with "EMP_" and by prefixing Session Name values with "EMP:". Avoiding use of names like these will ensure that the package can co-exist with other packages in your schema.

2.1 Record Types

The EmailPlus package introduces 7 new stateless record types into the schema to which it is applied. These are:

- EmailPlusConfig:
Contains the main controls for configuring the package.
- EmailPlusTemplate:
Defines the templates for the content of an email.

- **EmailPlusRule:**
Defines the conditions under which an email should be generated.
- **EmailPlusSiteConfig:**
Defines information about the SMTP and Web servers at each site in a ClearQuest MultiSite installation or the single site in a non-replicated environment.
- **EmailPlusAction:**
Is used by an internal caching mechanism to minimize the impact on performance of rule evaluation. There is no direct interaction with this record type.
- **EmailPlusRuleSubscription:**
Permits individual users to personally subscribe to notifications generated by individual rules.
- **EmailPlusPostOffice**
Allows requests for Email sending to be written to the ClearQuest database which the relay service then processes.

The relationship between these record types is shown in Figure 1.

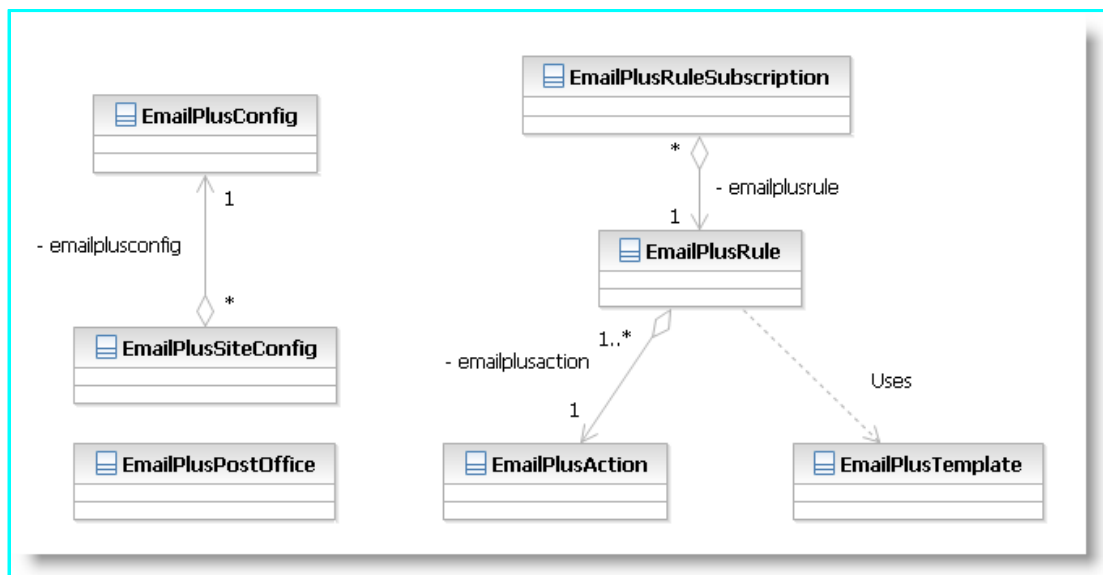


Figure 1. EmailPlus Record Relationships

2.1.1 Replicated versus Non-Replicated Environments

The EmailPlus package is designed to work in both replicated and non-replicated ClearQuest environments. In this context, 'Site' will refer to either the ClearQuest MultiSite replica name or in a non-replicated environment <local> i.e. the only site.

There is also another special value of Site, that of <default>. This value applies in the absence of a specific site definition.

Configuring Email Addresses and Clients for ClearQuest Users

EmailPlus relies on ClearQuest users to have the correct email address defined in their account profile. This can be done by any user with ClearQuest User administrator privileges. Also, all ClearQuest users can maintain their own email address via the ClearQuest web via the User Profile menu at the top right of the page. The ClearQuest users can also update their email address using the Eclipse Client when they are logged into a database as follows:

Tools -> Administration -> Change User Profile -> <connection profile>


The Email Options which can be defined in the ClearQuest clients are generally overridden by the settings defined in the EmailPlus configuration records. Therefore, EmailPlus does not require the ClearQuest client Email Options to be defined, however, an exception to this rule is if you use EmailPlus with the CQMail mail transport setting for one or more ClearQuest sites. In this case you will need to ensure that your ClearQuest clients do have Email notification enabled in the Email Options. In this case the Email Provider, Outgoing SMTP Server and Email Address defined in the Email Options will be overridden with the settings defined in the EmailPlus configuration and the email address defined in the ClearQuest users account profile.

Table 1 below illustrates when EmailPlus notifications will be sent based on the ClearQuest Client Email Notification setting status and the EmailPlus mail transport type:

Mail Transport Type	Email Notifications Enabled in ClearQuest Client	Email Notifications Disabled in ClearQuest Client
CQMail	YES	NO
Direct SMTP	YES	YES
SMTP Relay	YES	YES
PostOffice Relay	YES	YES

Table 1. Determining when EmailPlus notifications will be sent

2.2 EmailPlusConfig Record Type

There must be exactly one EmailPlusConfig record. Its purpose is to control the operation of the package and to record some configuration parameters. Refer to Figure 2. The generation of email notifications by the package is controlled globally by the EmailPlus Active checkbox . When unchecked, this will stop EmailPlus generating any emails. This is useful when, for example, some maintenance activity is being conducted on the database that may otherwise cause many unwanted emails to be generated.

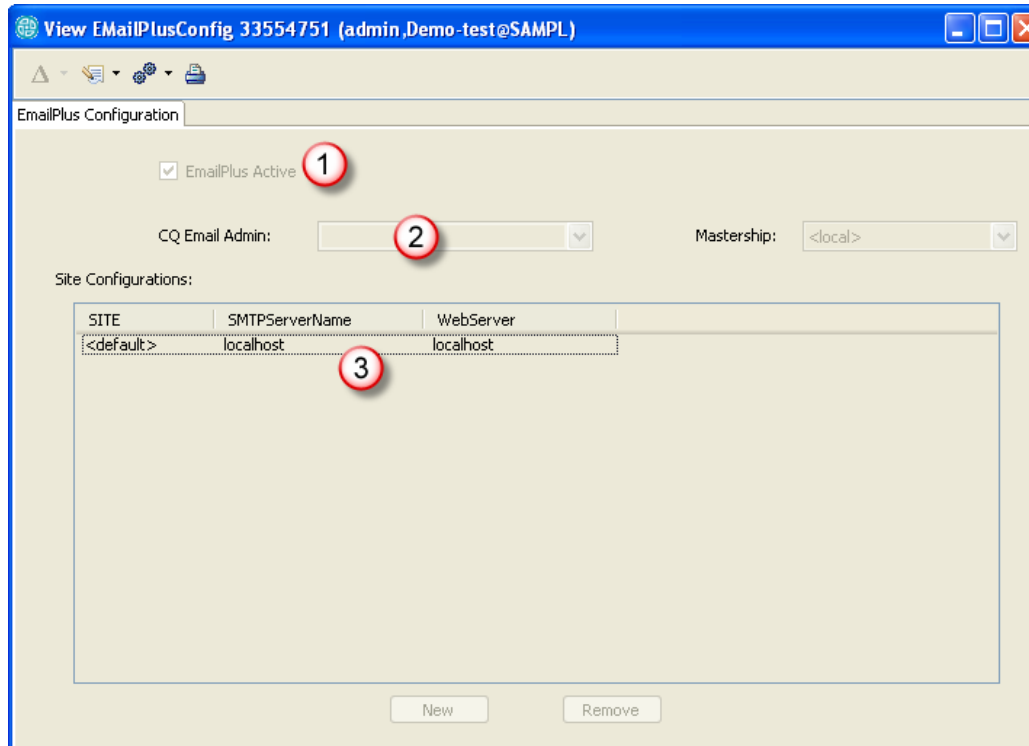


Figure 2. EmailPlus Configuration Record

In the event of any problems sending emails, EmailPlus will generate a notification of the fault and sends it to the ClearQuest user identified as the *CQ Email Admin* (2). It is important that this user has their email address defined in the ClearQuest user database for this to work. If this field is left blank, no notifications will be sent about email sending failures. The level of detail included in any notifications depends upon the transport mechanism used and what it reports if anything.

The site configurations field is a list of references to EmailPlusSiteConfig records that define information for each site in the ClearQuest installation.

Note: When a site configuration record is removed, it is also deleted. Orphaned records are not allowed.

2.3 EmailPlusSiteConfig Record Type

The EmailPlusSiteConfig records define information about the SMTP and ClearQuest web servers available at a given site. There must be at least one EmailPlusSiteConfig record. In a nonreplicated environment that is all that is needed. In a MultiSite environment, there can be up to one record per MultiSite replica. This record type has two tabs that are described in the following sections.

2.3.1 EmailPlusSiteConfig – SMTP Tab

This tab deals with the configuration of SMTP mail server details. Refer to Figure 3. In box (1) select the name of the replica to which the configuration applies. If this is a non-replicated ClearQuest installation, select <local>. You may also select the <default> site.

In this case, the <default> definitions will apply if a specific record for any given site doesn't exist.

In the area identified by **2**, enter the credentials for the SMTP server that is accessible to this site.

The SMTP Reply-to address, if set, will be the address used by email clients to respond to generated emails. Each site may have a different reply to address.

Note: Although a "reply-to" address may be specified, it is only supported by the transport types not based on the ClearQuest mail object. Custom transport types may support this.

The field identified by **3** allows you to select the transport type that will be used at that site. Features offered by EmailPlus (such as the support for rich text in emails) vary depending upon the transport type selected. Refer to section 2.3.1 for a description of the transport types supported by EmailPlus.

The field identified by **5** is only used in PostOffice Relay mode. It identifies the site name of the MultiSite replica where the Post office deliveries will be undertaken for this site. This is necessary because the records can only be deleted on the site where they are mastered. Having a remote site delivery will delay emails by up to one replication cycle period. If delivery time is of the essence, all Post Office deliveries should be performed locally. This means having a relay service running at each replica you care about. In non-replicated environments, this should be set to "<local>".

Finally, the fields identified in **4** are not used by EmailPlus directly, but are made available to any custom transport type defined.

View EmailPlusSiteConfig

SMTP Config | CQWeb Config

Site: <default> Mastership: <local>

Site that the following information applies to. Use <default> for sites that don't have explicit values.

SMTP Server Name: localhost

SMTP Port: 25 Email Transport Type: SMTP Direct

SMTP From Address: clearQuest@emp.com

SMTP Reply to Address: clearQuest@emp.com

NOTE: When using SMTP RELAY as the transport type, the SMTP Port and the Server Name refer to the RELAY service.

Post Office Mastership Site:

Authentication:

☐ SMTP is Authenticating

Credentials:

SMTP User ID:

SMTP Password:

☐ SMTP uses Secure Password Authentication

NOTE: This information is collected for use by CUSTOM solutions to use.
None of the Built in solutions use authenticating SMTP.

OK Cancel Print Record Actions

Figure 3. EmailPlus Site Configuration Record – SMTP Config Tab

2.3.1.1 Email Transport Types

The EmailPlus package offers a number of different transport types to suit different needs within the enterprise. You may select different transport types for different sites in a ClearQuest MultiSite environment. Figure 4 shows the possible email transport types.

Email Transport Type:

SMTP Relay

CQMail

Custom

SMTP Direct

SMTP Relay

PostOffice Relay

Figure 4. Email Transport Types

CQMail:

This transport type uses the standard ClearQuest mail object to transport emails. If this transport type is selected, the server details are taken from the client configuration as normal and not taken from this EmailPlus record. The CQMail transport type has all the limitations experienced with the standard Email Rules package in regard to HTML Content, Mail headers and attachments. However it is the only way to use a MAPI mail transport.

Custom:

This transport type gives you all the benefits of the rich content and rules of the EmailPlus package but leaves the actual relay of messages to a custom solution provided by you.

```
sub EMP_SendMail_CUSTOM (
    my $fromAddr = shift; # String
    my $toList_ref = shift; # Reference to an array
    my $ccList_ref = shift; # Reference to an array
    my $bccList_ref = shift; # Reference to an array
    my $subject = shift; # String
    my $body = shift; # String
    my $isHTMLContent = shift; # Boolean flag
    my $ccActioner = shift; # Boolean flag
    my $attachmentList_ref = shift; # Reference to an array
    my $mailHeaders_ref = shift; # Reference to an array

    # This is a customisable template to deliver mail in whatever way
    # you see fit. It is only called for sites configured to use the
    # 'CUSTOM' email transport type. Its up to you to get your mail there

    # You can retrieve information about the local SMTP server by calling the function
    # EMP_LookupSMTPServerAttribute(AttName)
    # with one of these strings as AttName to request that item of information
    # "SITE", "SMTPServerName", "SMTPPort", "isSMTPAuthenticating", "SMTPUserID"
    # "SMTPPassword", "isSMTPSecurePw" "SMTPReturnAddress", "SMTPFromAddress",
    # "emailTransportType"
    EMP_DebugOut("START - Deliver mail using custom format");
    # EMP_WebSafeMsgBox("REMEMBER to customise the EMP_SendMail_CUSTOM routine to your specific
    EMP_DebugOut("REMEMBER to customise the EMP_SendMail_CUSTOM routine to your specification
    EMP_DebugOut("END");
```

Figure 5. Custom Email Transport Routine

In ClearQuest Designer, there is a function in the PERL Global Script 'EMP_Customisable' called **EMP_SendMail_CUSTOM** that you can edit to provide the transport functionality you want (see Figure 5). If you do not customize this function, the custom transport will only display a warning message when running EmailPlus in debug mode (see Appendix G).

There is also a function in the PERL Global Script 'EMP_Customisable' called **EMP_SiteConfig_CUSTOM_SetFormFieldRequiredness** which will allow you to customize the field requiredness of the fields on the EmailPlusSiteConfig record for the custom email transport type (see Figure 6). By default, all of the fields are optional, but you could modify this to make certain fields mandatory or read only.

```
sub EMP_SiteConfig_CUSTOM_SetFormFieldRequiredness (
    # Modify the values of the field requiredness for each fields on the EmailPlusSiteConfig record form

    $entity->SetFieldRequirednessForCurrentAction("SMTPServerName", $CQPerlExt::CQ_OPTIONAL);
    $entity->SetFieldRequirednessForCurrentAction("SMTPPort", $CQPerlExt::CQ_OPTIONAL);
    $entity->SetFieldRequirednessForCurrentAction("SMTPFromAddress", $CQPerlExt::CQ_OPTIONAL);
    $entity->SetFieldRequirednessForCurrentAction("SMTPReturnAddress", $CQPerlExt::CQ_OPTIONAL);
    $entity->SetFieldRequirednessForCurrentAction("isSMTPAuthenticating", $CQPerlExt::CQ_OPTIONAL);
    $entity->SetFieldRequirednessForCurrentAction("PostOfficeMastershipSite", $CQPerlExt::CQ_READONLY);
```

Figure 6. Custom SetFormFieldRequiredness

SMTP Direct:

This transport type makes use of a direct connection to the SMTP server that allows for a more robust email transport, but only via SMTP. This transport type allows the package to send emails that have HTML body parts that may also contain embedded images and additionally allows the use of attachments and SMTP mail headers.

SMTP Relay:

This is essentially the same as SMTP Direct except that the message components are sent as an XML data stream to a relay component which in turn receives the XML data and converts it to an email which is then delivered to your SMTP server. This allows the package to operate in environments where the SMTP server will only accept requests from named IP addresses. See section 7 for more information about the relay service.

PostOffice Relay:

This transport type records the details of Emails required as records in the ClearQuest database. Each email required creates one record of type EmailPlusPostOffice. These records contain everything necessary for the relay service component to retrieve and generate the emails. See section 7 for more information about the relay service.

2.3.2 EmailPlusSiteConfig - CQ Web Tab

The ClearQuest Web Config tab defines information about the ClearQuest web server (if any) at the site. These credentials are used in constructing click through links in the email message body to take you to a specific record, chart, query or report.

The server name and port number are defined in fields show identified by ① in Figure 7. If the server name is left blank, then it is assumed there is no web server at this site. The port number can be left blank if you are using the default port number.

You can pick the protocol ② you use to talk to your web server, either http:// or https://. Your ClearQuest web server should have been configured appropriately to use https://

If you wish, you can specify a ClearQuest login ID and password ③. The URLs generated will then take the user straight to the appropriate item. However you should be aware that using this facility is a potential security risk as the user ID and password will appear as plain text in each URL. Also, the user will be able to perform whatever actions that ClearQuest ID is permitted to do. If you leave these fields blank, the user who clicks on the link will have to provide their own ClearQuest credentials if they do not already have an active ClearQuest web session.

View EmailPlusSiteConfig <default> (admin.Demo-test@SAMPL)

SMTP Config | CQWeb Config

Web Server: localhost 1

Web Server Port: 80

Web Server Protocol: http 2

Web User ID: 3

Web Password:

These credentials are used to construct a URL to a ClearQuest record in a generated email (if specified in an EmailPlus rule).
The server credentials used will be either:

- a. The web server for the site that matches the mastership of the recipients user record, or
- b. The web server specified in the default site record if there is no record that matches the recipients user record mastership.

Figure 7. EmailPlus Site Configuration Record - <stet>Config Tab

2.4 EmailPlusRule Record Type

The EmailPlusRule record type is at the heart of the package. Its purpose is to define:

- The rule conditions under which an email notification will be generated,
- Whether or not self subscription to the rule is permitted,
- The email addresses to whom notifications will be sent, and
- The EmailPlus template that will be used to generate the notification content.

The EmailPlusRule record type is comprised of 6 tabs that are fully described in detail in section 4.

Note: In ClearQuest MultiSite environments, it is a requirement of the package that all rules relating to a single record type and its associated EmailPlusAction record are mastered at the same site. This is required since each time an EmailPlusRule is created or edited, the EmailPlusAction record is updated and this can only be achieved if they are all mastered on the same site. The package goes some way to ensure this.

2.5 EmailPlusTemplate Record Type

The EmailPlusTemplate record type is used to define the content and layout of an email. A template can be used by multiple rules for the same record type if required. The EmailPlusTemplate record type comprises five tabs which are described in the following sub-sections.

2.5.1 EmailPlusTemplate – Template Definition Tab

The template definition tab specifies some basic information about the template. It defines things like its name, kind and a description of what it is used for.

To create a template, firstly select the record type that the template is to be defined for. See ① in Figure 8. Until this is done, it will not be possible to fill in some other parts of the template record.

The Template Name ② is an arbitrary title for the template that identifies its purpose but it must be unique in the population of Templates for a given record type.

The Description ④ allows a fuller description of what the template is used for.

The Template Kind Field ③ allows the administrator to define the type of template to be created. There are three kinds to choose from:

- **Special Interest:**

This type of template is used by the EmailPlus package to generate notifications to individuals who have personally subscribed to notifications about changes to a *specific record(s)*. There can be only one of this kind of template per record type. Whenever *any* change occurs to a record the user is subscribed to, they will receive a notification of the change. Special Interest templates are never explicitly associated with an EmailPlusRule. The rule is implicit in the EmailPlus package.

Note: Users may also subscribe to an *EmailPlus Rule* in which case they will be notified whenever the rule fires. See section 4.6.

- **Ownership:**

This type of template is used to notify a change of ownership. Whenever any of the nominated ownership fields (See section 6.4) are changed, both the old owner and the new owner are notified of the change by an email generated from this template. Like the Special Interest template type, there may only be one of these per record type. Ownership templates are never explicitly associated with an EmailPlusRule. The rule is implicit in the package.

- **Rule:**

The third and final template type is rule. Templates of this type need to be associated with one or more rules. When a rule is triggered an email is generated using the template associated with it.

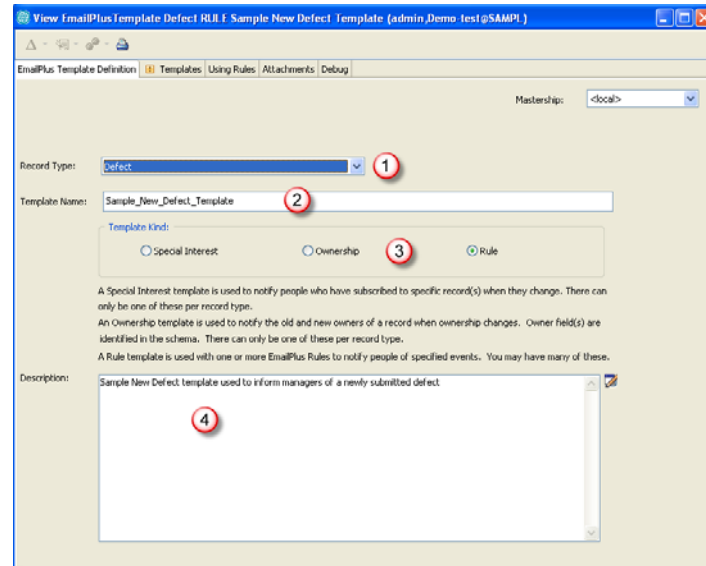


Figure 8. EmailPlus Template Record - EmailPlus Template Definition Tab

2.5.2 EmailPlusTemplate – Templates Tab

This tab (see Figure 9) contains the definitions to be used for the mail subject (header) ¹ and mail body ³ parts of the message. These template fields may contain plain text and special substitution tags that get expanded to the values they represent when the template is applied. Optionally, the administrator can also specify that the body part be treated as HTML ². In that case the body may contain any valid HTML tags that would normally appear between <BODY> and </BODY> tags. This facility is dependent on the email transport type selected and only works with the SMTP Direct, SMTP Relay and Post-office Relay options or the custom one if the custom transport so permits.

For template bodies that are specified as HTML, when you need to force a newline in the rendered HTML place a
 tag into the HTML source as shown in Figure 9.

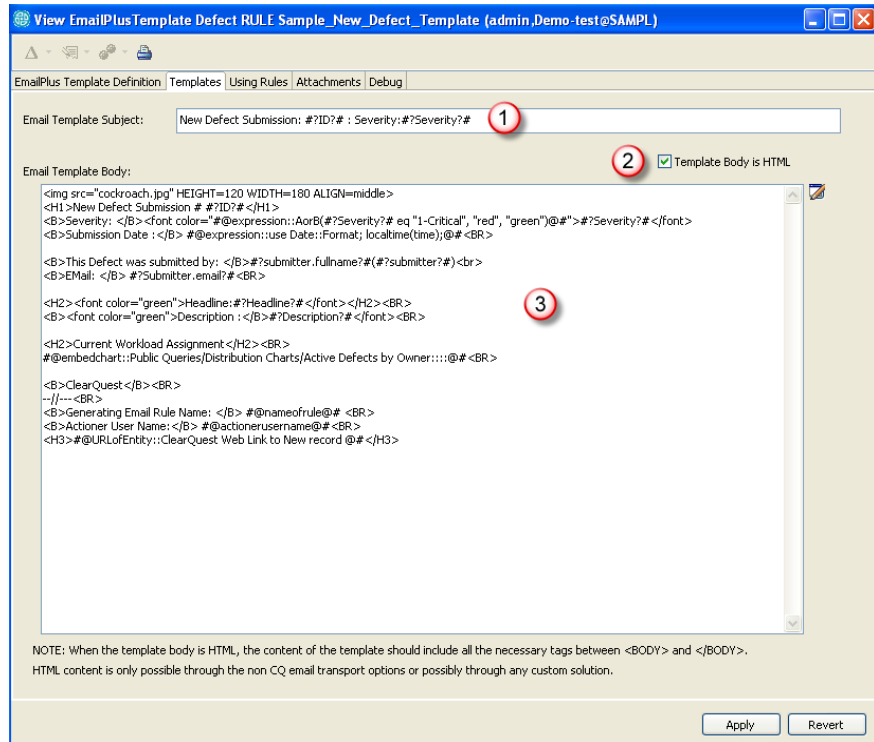


Figure 9. EmailPlus Template Record - EmailPlus Templates Tab

The list of template substitution markers is extensive and documented in section 3.

2.5.3 EmailPlusTemplate – Using Rules Tab

The Using Rules tab (see Figure 10) provides a list of all the rules that are currently associated with the template. So you can see at a glance, what rules you will be affecting if you change anything in the template. This should make it easier to decide whether or not to create a new template or to modify the existing one.

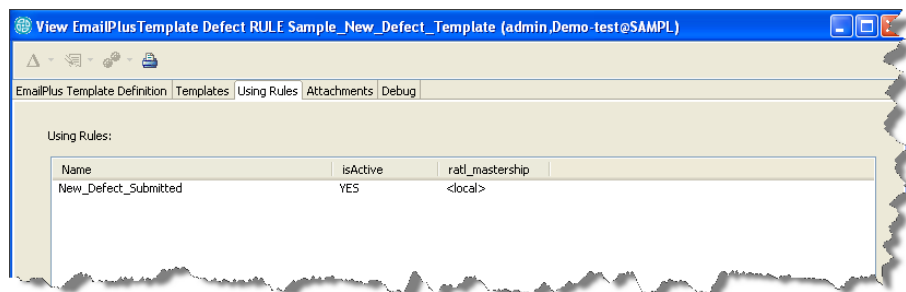


Figure 10. EmailPlus Template Record - Using Rules Tab

2.5.4 EmailPlusTemplate – Attachments Tab

EmailPlus does not use the attachments package and because of this will not make use of changes in the storage model that newer versions of the attachments package may provide. Template attachments are always stored in the ClearQuest database.

The Attachments Tab (see Figure 11) allows the administrator to

- Attach graphic images (or other files) that are to be unconditionally attached to any generated emails created using this template. These attachments are used for embedded graphics in the template. So for example if you wanted a picture of a bug in your defect emails, you would attach the graphic for it here and then refer to in an HTML tag. See section 3.2 for details on how to do this.
- Specify any mail headers. This allows you to do things like specify a priority for generated emails. Mail headers consist of two components separated by "::-". The two parts are Name and value. So in the example, 'x-Priority' is the mail header name and '1-high' is the value of the mail header. Mail headers are only supported by the SMTP Direct, SMTP Relay and Post-Office relay mail transports provided that the ClearQuest mail component is not used. The custom transport may also support this but this is implementation dependent.

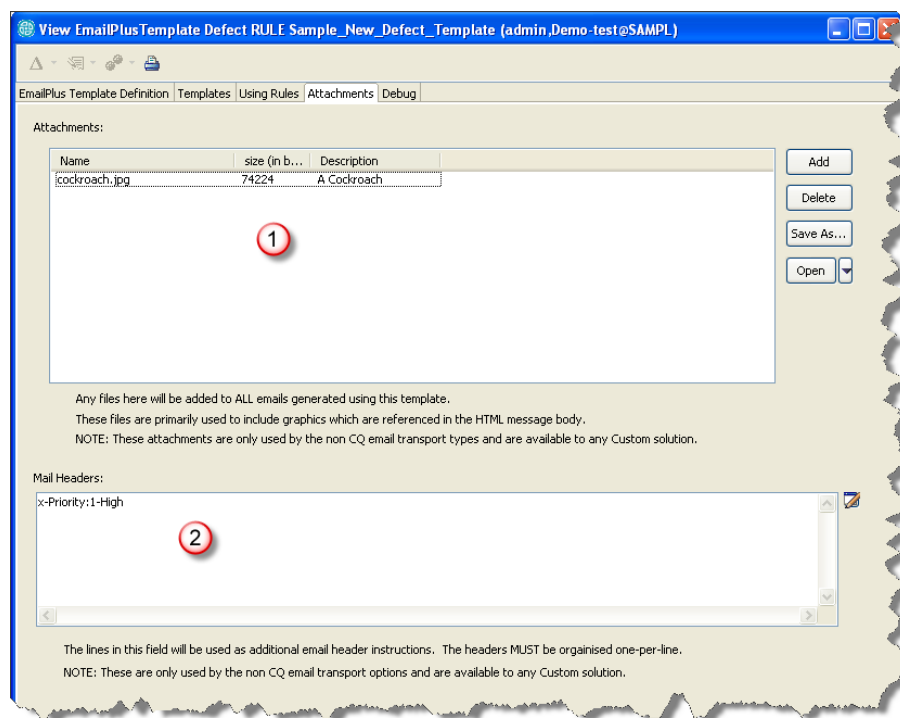


Figure 11. EmailPlus Template Record - Attachments Tab

For more information on Email Headers see [RFC 2822 - Internet Message Format](#).

2.6 EmailPlusAction Record Type

The EmailPlusAction records are created automatically by the EmailPlus package whenever the first EmailPlus Rule is created for a given record type. Their purpose is to cache all of the rules so that when the rules have to be evaluated, it minimizes the performance

impact. The EmailPlusAction record type is an internal only record type and has no user serviceable parts. If for some reason you believe that the cache may be out of date, just amend any rule for the given record type and the cache record will be rebuilt.

Note: It only rebuilds the cache record, it doesn't update the in memory caches on any other active clients. To update these caches, users will need to log out and then back in again.

2.7 EmailPlusRuleSubscription Record Type

The EmailPlusRuleSubscription record type (Figure 12) exists to allow individual users to register an interest in receiving notifications generated by the associated EmailPlus-Rule. The subscription record is a separate entity in order that in a ClearQuest MultiSite environment there can be an editable subscription record at each replica the administrator wants the facility to be available from. If no EmailPlusRuleSubscription records are created for a rule, then no subscription is possible for that rule.

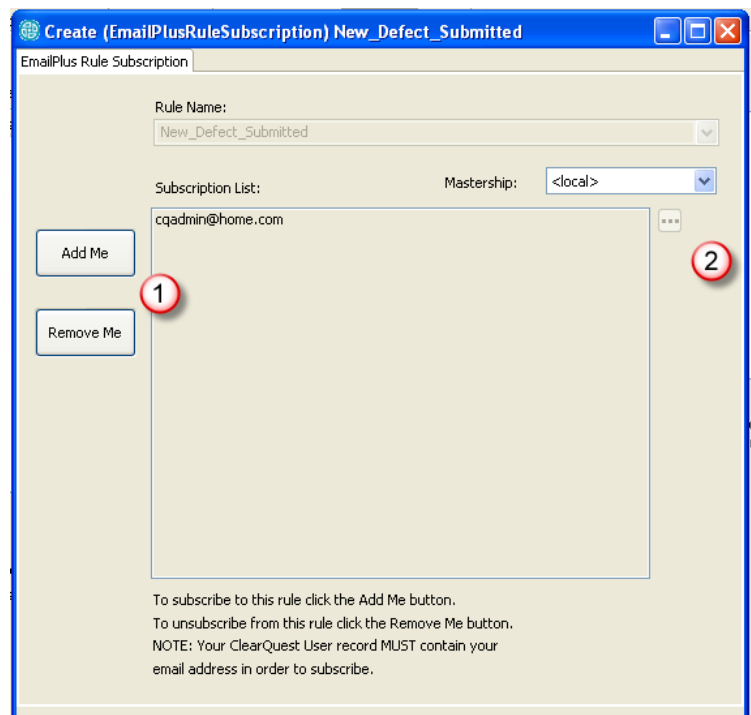


Figure 12. EmailPlus Configuration Record

Note: EmailPlus has to query the database to find the subscription records and their members but it only does this for rules that are evaluated true. These are *not* cached so new subscriptions take effect almost immediately although there may be some delay in MultiSite configurations until remote records are replicated.

Users subscribe to a rule by locating the subscription record for the rule they wish to subscribe to in their local replica. Users can add themselves to the subscription list by clicking the **Add Me** button (see 1) or remove themselves by clicking **Remove Me**. If needed, the EmailPlus administrator can also restrict who can subscribe to a rule. See section 4.6.

2.8 EmailPlusPostOffice Record Type

When the PostOffice Relay transport type has been selected, EmailPlusPostOffice records are created automatically when there is a need to send an email. The records are processed and deleted by an instance of the relay service that is described in section 7.

The EmailPlusPostOffice record type is an internal only record type and has no user serviceable parts although if there are problems caused by an individual record, it could be deleted with a script using a login ID that is a member of the EmailPlusAdmins group.

If you are not using the PostOffice Relay transport type, then there should be no records of this type.

3 Working with EmailPlus Templates

Email templates consist of boilerplate text that has special substitution tags embedded in it. These tags are replaced by the value that they represent when the template is applied. The boilerplate can contain plain text or html.

Three types of tag are supported by EmailPlus:

- Current Value Field Tags
- Old Value Field Tags and
- Meta Tags

All tags are enclosed by special character sequences that define the tag type as shown in Table 1.

Opening Sequence	Closing Sequence	Tag Type	Example
#?	?#	Current field value. Any field defined in the ClearQuest record.	#?state?#
#%	%#	Old Field Value. Any field defined in the ClearQuest record	##state##
#@	@#	Meta Tag Name (See section 3.1.	#@URLofEntity@#

Table 2. EmailPlus Tag Definitions

These tag substitutions make EmailPlus extremely powerful when defining the content of a generated email. EmailPlus is not limited to data contained in the record being modified. Information from anywhere in the ClearQuest environment can be used.

3.1 Template Meta Tags

There are many Meta tags built into the EmailPlus package. This Meta tag facility is also extensible by the ClearQuest administrator. Custom Meta Tags can be added by modifying the schema (See Appendix A for details).

Table 2 describes the built-in Meta tags supported by Email Plus. Some of these tags have parameters. Some parameters are optional or have default values. Optional parameters are enclosed in { } characters which are not part of the parameter syntax. Meta tags should be defined on a single line although in Table 2 the Meta Tag Name column may display them over multiple lines.

Meta Tag Name	Purpose
URLofEntity {::<linktext>}	<p>Replaces the tag with a URL generated to take the user directly to the record that caused the email rule to fire. The web server credentials are those defined in the Email-PlusSiteConfig record for the site where the record is mastered. If no site record exists for the replica, <default> is used. If no web servers are defined then no URL can be generated.</p> <p><linktext> is an optional string containing the text users see in HTML body parts, instead of the URL itself.</p> <p>Example:</p> <p>#@URLofEntity::Click Here To Visit Record@#</p> <p>Would insert the URL into the email and display it as:</p> <p>Click Here To Visit Record</p>
URLofQuery::<queryname> {::<linktext>}	<p>Replaces the tag with a URL of the named query.</p> <p><queryname> is the name of a ClearQuest query. This should be a Public query unless it can be guaranteed that every user has the same query in their Personal Queries Folder.</p> <p><linktext> is an optional string containing the text users see in HTML body parts instead of the URL itself.</p> <p>Example:</p> <p>#@URLofQuery::Public Queries/All Defects::Click Here To Run Query@#</p> <p>Would insert the URL into the email and display it as:</p> <p>Click Here to Run Query</p>
URLofChart::<chartname> {::<linktext>}	<p>Replaces the tag with a URL of the named chart.</p>

	<p><chartname> is the name of a ClearQuest chart. This should be a Public chart unless it can be guaranteed that every user has the same query in their Personal Queries Folder.</p> <p><linktext> is an optional string containing the text users see in HTML body parts instead of the URL itself.</p> <p>Example:</p> <p>#@URLofChart::Public Queries/Defect Distribution::Click Here To Display Chart@#</p> <p>Would insert the URL into the email and display it as:</p> <p>Click Here To Display Chart</p>
URLofReport::<reportname> {::<linktext>}	<p>Replaces the tag with a URL of the named report.</p> <p><reportname> is the name of a ClearQuest report. This should be a Public report unless it can be guaranteed that every user has the same query in their Personal Queries Folder.</p> <p><linktext> is an optional string containing the text users see in HTML body parts instead of the URL itself.</p> <p>Example:</p> <p>#@URLofReport::Public Queries/Active Defects::Click Here To Display Report@#</p> <p>Would insert the URL into the email and display it as:</p> <p>Click Here To Display Report</p>
ActionerUserName	<p>Replaces the tag with the ClearQuest user name of the person who performed the action that caused the rule to fire.</p> <p>Example:</p> <p>#@ActionerUserName@#</p> <p>If the user name “owenja” had caused the EmailPlus rule to fire then this user name would be inserted into the email.</p>
NameofUserInField::<field>	<p>Replaces the tag with the full name of the user whose ClearQuest login ID is held in the field <field>.</p> <p>Example:</p>

	<p>#@NameofUserInField::owner@#</p> <p>If the ClearQuest login ID in the owner field is "owenja" then their full name "John A. Owen" would be inserted into the email.</p>
NameOfRule	<p>Replaces the tag with the name of the rule that fired and expanded the template.</p> <p>Example:</p> <p>#@NameofRule@#</p> <p>If the rule that fired was "RequestHas-ChangedState" then this text would be inserted into the email.</p>
Expression::<expression>	<p>This is perhaps the most powerful Meta tag of all. The operand <expression> may be any valid PERL expression. This expression may include any PERL built-in function or any function defined in the global scripts section of the schema to which the package is applied. The value returned by the evaluation of the expression replaces the tag. Expressions can include embedded field expressions which are described further in section 3.1.1..</p> <p>Example:</p> <pre>#@Expression:: use Date::Format;use Date::Parse; my \$template = "%d-%m-%Y %H:%M:%S"; time2str(\$template, str2time(#?SubmitDate?#, "UTC"));@#</pre> <p>This expression takes the current value of the field SubmitDate and formats it using Perl. So if SubmitDate contains the date/time 23 February 2009 at 5:40am this would be formatted by the expression and the following text would be inserted into the email:</p> <p>23-02-2009 05:40:00</p>
AttachmentByName::<attFieldName>{::<Pattern> ¹ }	<p>Inserts attachments into the email from the attachments field called <attFieldName>.</p> <p>Optionally, a regular expression² can be provided to select which attachments are added.</p>

¹ Note: If the target schema uses a version of the ClearQuest attachments package that uses an external storage model, this feature will not work. However it may be possible to include a link to the external storage site. See Attachments package documentation for more information on where the link is stored.

	<p>If no pattern is given all will match. The filenames (not the paths) are used in the pattern match.</p> <p>Example:</p> <pre>#@ AttachmentByName::Attachments::^Spec.*@#</pre> <p>Matches all attachments whose file name starts with "Spec" and attaches these files to the email.</p>
AttachmentByDescription::<attFieldName>{::<Pattern> ² }	<p>This is the same as for AttachmentByName except the pattern match is against the Attachment description instead of the filename.</p>
EmbedChart::Public Queries/Chart { {::<height>> } {::<Width>> } }	<p>Inserts a ClearQuest chart into the email. The first parameter is the fully qualified name of the chart.</p> <p>Note: This should be a public chart to ensure visibility for all users.</p> <p>You can optionally specify the size of the chart in pixels. If height and / or width are omitted the default size of 500 x 800 pixels is used.</p> <p>Example:</p> <pre>#@ EmbedChart::Public Queries/Chart::300:400@#</pre> <p>Embeds the chart "Public Queries/Chart" with a size of 300 pixels high by 400 pixels wide into the email.</p>
Other{::<your parameters>}	<p>This is the extensibility facility. You can define your own Meta tags in the schema by editing the function EMP_SubstituteCustomMetaTag in the Global script EMP_Customisable. See Appendix A for further details.</p>

Table 3. EmailPlus Meta Tags

3.1.1 Working with Embedded Field Expansions in Meta tags

In this revision of the package, it became possible to embed current and old-field value tags within the expression tag itself. Whilst in earlier version this was possible to formulate the expression, it wasn't possible to get a template to validate correctly if you did this.

² A full list of regular expression Meta tags can be found in Appendix B.

EmailPlus now recognizes an expansion of an old or current field value as a special case. Rather than just returning the value in the field as a straight piece of text, it returns the value as a properly formatted expression so that when provided as a parameter in the expression, the expression is syntactically correct and thus the template can validate.

As an example, lets assume that you wanted to display the submit date from your defect record, but you didn't want the default date format, you wanted to display it in a more readable form.

So here is our example expression meta tag:

```
#@expression::use Date::Format;use Date::Parse; ctime( str2time( #?SubmitDate?# ) );@#
```

This is expanded to

```
#@expression::use Date::Format;use Date::Parse; ctime( str2time( "2008-07-15 10:45:18" ));@#
```

This is a syntactically valid expression; it returns a date object. When the above expression is evaluated, it returns the following:

Mon Jul 15 10:45:18 2009

This is what is substituted into the output message.

Table 4 lists the kinds of data types returned when an old or current field value is expanded within a meta tag.

Field Type	Expression Returned
INT	Integer For example, 54
DATE_TIME	24 hour date string "YYYY-MM-DD hh:mm:ss" For example "2009-10-20 15:23:04"
SHORT_STRING, MULTILINE_STRING, ID, STATE, RECORDTYPE, DBID, REFERENCE & any other types not specified	Quoted string value For example "some text" Quotes embedded in the original field value will be properly dealt with to yield a syntactically correct string value
REFERENCE_LIST	A reference to an array of quoted values For example: ["value 1", "value 2"]

Table 4. EmailPlus data types returned in meta tags

3.2 Embedding Images in an HTML Body Template

To embed an image in the body of an HTML email message body, insert a tag like this one in the body template:

```

```

Store the image file for image.gif in the attachments field of the template (see section 2.5.4). It is important that the image name you specify here matches the filename of an attachment to the template. Attachments with one of the following extensions can be used as embedded graphics:

.jpg, .jpeg, .png, .gif

All files found in the attachments field are added to the body regardless of whether they are used in an tag. The treatment of additional files may depend on the email client. Some may treat them as simple attachments while others may display them at the end (or beginning) of the email.

Don't use a pathname with the *img* tag.

4 Working with EmailPlus Rules

The EmailPlus Rule record type provides the EmailPlus package with a rich environment for defining conditions under which email notifications are generated and who will receive copies of any emails generated by the rule. A correctly defined EmailPlus rule will include:

- An initial rule definition comprising
 - A name to uniquely identify the rule by,
 - The record type to which the rule applies,
 - The name of the EmailPlus template which will be used to construct the email header and body,
- The definition of the rule conditions which will need to be met in order for the rule to fire, and
- Details of the users or subscribers who will receive the email when the rule fires.

The rule conditions are constructed from one or more rule phrases. These phrases equate to logical expressions which evaluate to TRUE or FALSE when a change is made to the record type to which the rule applies. Example phrases are:

Project IS IN "Project A", "Project B", "Project C"

Priority > 1

Action performed on record is "Complete"

These phrases are defined on the EmailPlus Rule, Actions / States and Advance Rule tabs. Rule phrases are linked to each other using an AND/OR logical operator to create the overall rule conditions. Building upon the example above a rule condition might be:

Project IS IN “Project A”, “Project B”, “Project C”
AND Priority > 1
OR Action performed on record is “Complete”

In this example, the rule will only fire if the project field contains one of the values “Project A”, “Project B” or “Project C” and either the priority is greater than 1 or the complete action has been executed on the record. In this case the rule conditions have evaluated to TRUE. If one of the above conditions hadn’t been met (say because the project was called “Install new GUI”) then the rule conditions would have evaluated to FALSE.

The rule will only fire if the rule conditions evaluate to TRUE.

4.1 Initial Rule Definition

When a new rule is created, three things must be established before anything else meaningful is possible:

- First, the type of record that the rule applies to needs to be selected using the drop-down (see ① in Figure 13). This will allow EmailPlus to determine the characteristics of the record associated with the rule.
- Then the template associated with the rule needs to be specified ②. The template defines the structure and content of the notification message.
- Finally, the rule needs to be given a name ③. The name is any arbitrary text that adequately expresses the purpose of the rule.

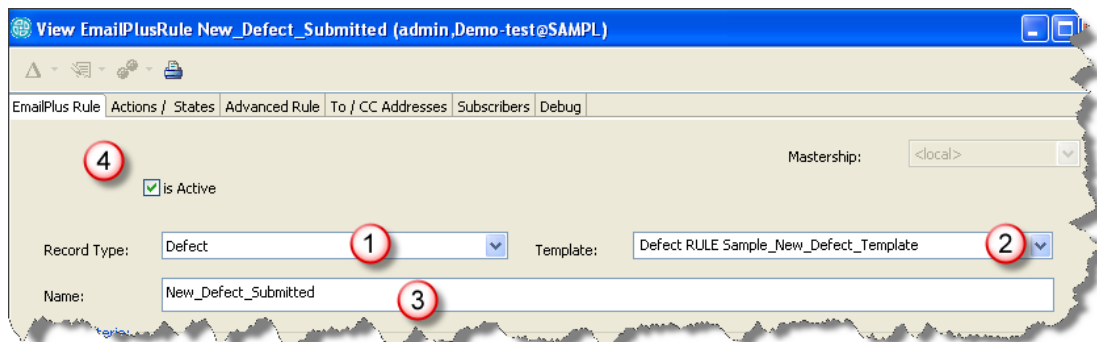


Figure 13. EmailPlus Rule Record – Initial Definition

Note: Individual EmailPlus rules can be turned on and off using the “is active” checkbox ④.

Once these basic pieces of information have been provided, the next step is to provide the definition of the rule itself using one or more rule phrases. The rule conditions can use many different components which are described in the following sections.

4.2 Defining a Rule Phrase Using Value Criteria

The value criteria or simple expression fields on the EmailPlus Rule tab allow you to create simple conditions around the values in fields (Figure 14). Up to 5 groups of value criteria can be provided. These will probably be enough to cover most simple conditions.

Unlike the out-of-the-box Email Rules package, EmailPlus value criteria *does not* use queries and therefore value criteria provide an efficient way to perform comparison operations.

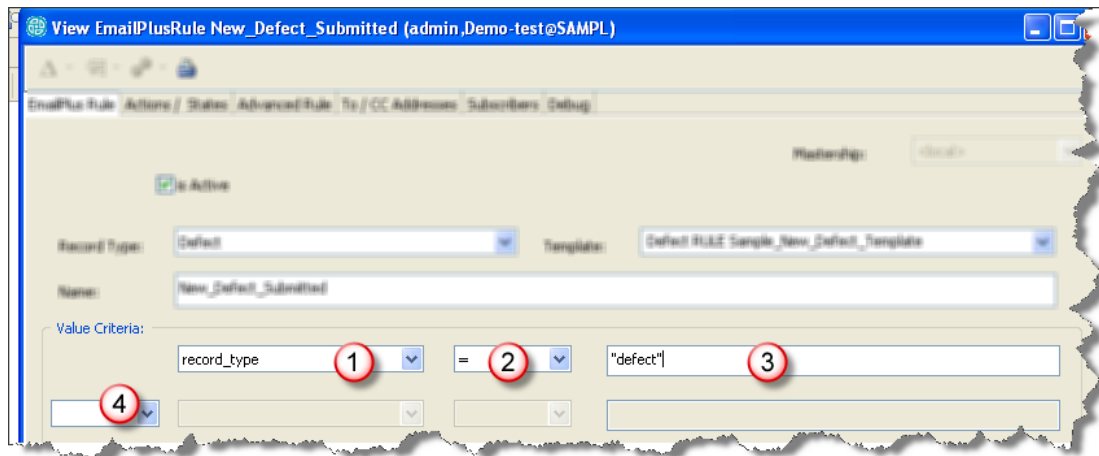


Figure 14. EmailPlus Rule Record - Specifying Value Criteria

The value criteria are made up of three parts:

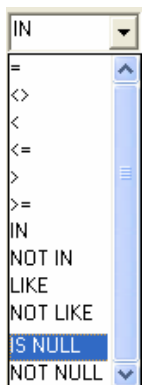
< Field Name Selection > < Comparison Operator > < Comparison Value >

Together these parts equate to a logical expression which evaluates to TRUE or FALSE. Examples of value criteria are:

ProjectName	IS LIKE	"Proj"
Priority	>=	1
Owner	NOT NULL	
ProjectName	IN	"Project A", "Project B", "Project C"

These three parts are described in further detail below.

- <Field name Selection> ①
Select one of the field names for the record type the rule applies to.
- <Comparison operator> ②
Select from the list of the supported operators.



=	(equals)
<>	(not equals)
<	(less than)
<=	(less than or equal)
>	(greater than)
>=	(greater than or equal)

Table 1 lists other comparison operators. Database administrators will recognize that these are similar to comparison operators provided in SQL.

Comparison Operator	Description
IN	<p>The value criteria will return TRUE if the value of the field provided in the <Field Name Selection> is one of the values provided in the <Comparison Value>.</p> <p>Example:</p> <p>Field Name Selection = <i>Project</i> Comparison Operator = <i>IN</i> Comparison Value = "<i>Project A</i>", "<i>Project B</i>", <i>Project C</i>"</p> <p>When the EmailPlus rule is evaluated if the value of the field Project is "Project A" then the value criteria will return TRUE. However, if the value of the field Project is "Project Z" (which isn't in the list of comparison values) then the value criteria will return FALSE.</p>
NOT IN	<p>This is similar to IN, except the value criteria will return TRUE if the value of the field provided in <Field Name Selection> is not one of the values provided in the list of <Comparison Value>.</p>
LIKE	<p>The value criteria will return TRUE if the value of the field provided in the <Field Name Selection> contains the text provide in <Comparison Value>, bearing in mind that string comparisons are case sensitive.</p> <p>Example:</p> <p>Field Name Selection = <i>Project</i> Comparison Operator = <i>LIKE</i> Literal Values = "<i>Proj</i>"</p> <p>When the EmailPlus rule is evaluated if the value of the field Project is "Project A" then the value criteria will return TRUE since "Project A" contains the text "Proj". However, if the value of Project is "New System Icon" then the value criteria will return FALSE since this project does not contain the text "Proj".</p>
NOT LIKE	<p>This is similar to LIKE, except the value criteria will return TRUE if the value of the field provided in <Field Name Selection> does not contain the value provided in <Comparison Value>.</p>
IS NULL	<p>The value criteria will return TRUE if the value of the field provided in the <Field Name Selection> is empty or has no value.</p> <p>Note: This comparison operator does not require the <Comparison Value> to be set.</p> <p>Example:</p>

	<p>Field Name Selection = <i>Project</i> Comparison Operator = <i>IS NULL</i> Literal Values not used</p> <p>When the EmailPlus rule is evaluated if the value of the field project is "" (empty) then the value criteria will return TRUE. However, if the value is not-empty (for example, "Project A") then the value criteria will return FALSE.</p>
NOT NULL	<p>This is similar to IS NULL, except the value criteria will return TRUE if field provided in <Field Name Selection> is not empty.</p>

Table 1. EmailPlus Rule Record - Additional Logical Operators Supported

- <Comparison Value(s)> ³ <Can't determine what this refers to>

This provides the value or list of values (when using the IN/NOT IN comparison operators) against which the value in the field will be compared using the comparison operator.

Comparison values do not need to be provided when using the comparison operators IS NULL and NOT NULL.

String values must be enclosed in quotation marks, for example "Project A".

Lists of values must be represented as a comma separated list, for example "Project A", "Project B", "Project C" for a list of strings or 1,2,3 for a list of numbers.

Comparison values can also be specified as a Perl Expression which returns a value which is of the same type as the field in the value criteria and which is also valid for the comparison operand used. The expression could call a routine defined in your schema, or a ClearQuest API call. So for example, to only send an email if the current user was the submitter, you could use the ClearQuest GetUserLoginName() API call to return the current user as shown in Figure 15.

The screenshot shows a web-based interface for configuring an EmailPlus rule. The 'Name' field is set to 'New_Email_Submitted'. The 'Value Criteria' section contains a dropdown menu with 'Submitter' selected, followed by an equals sign (=) in a dropdown, and a text box containing the Perl expression '\$entity->GetSession()->GetUserLoginName()'. Below this, there are additional dropdown menus for logical operators and other criteria, but they are not fully visible.

Figure 15. EmailPlus Rule Record - Example value criteria using a ClearQuest API call

You can also use this mechanism to add your own dynamic comparison values to the EmailPlus value criteria. For example, you could add your own function to the schema which returns date comparison values like YESTERDAY, LAST_WEEK, etc.

- When multiple value criteria conditions are in use, you can combine them by selecting the required AND/OR logical operator (see ⁴ in Figure 14).

While this facility is quite expressive, it's not quite as flexible as the ability to formulate more complex expressions, as it would be with a ClearQuest Query. However there is an advanced expression facility which is explained further in section 4.7.

4.3 Defining a Rule Phrase Using Field Changes

The field changes section of the EmailPlus Rule tab (Figure 16) allows the addition of a phrase to the rule condition which looks for fields which have changed, rather than the values they contain. This rule phrase comprises three parts:

< Phrase Operand > < Fields Operand > < Fields Changed >

Figure 16. EmailPlus Rule Record - Specifying Field Changes

< Phrase Operand >

If you wish to look for changed fields, first you must select the phrase operand (AND or OR) from the pick list (see 1 in Figure 16). This will combine the fields change phrase with any value criteria phrases defined previously.

Note: Even if there are no value criteria then AND/OR still needs to be selected in order to make the input fields associated with this rule phrase editable.

< Fields Operand >

You then have to decide what combination of field changes you are looking for by selecting the required fields operand from the list (see 2 in Figure 16):

ALL

All the fields listed in < Fields Changed > must have changed in order for this phrase to evaluate to TRUE.

ANY

At least one of the fields listed in < Fields Changed > must have changed in order for this phrase to evaluate TRUE.

NONE

None of the fields listed in < Fields Changed > must have changed in order for this phrase to evaluate to TRUE.

< Fields Changed >

Finally, you need to select the fields on which to base this rule phrase. To do this, click the multi-select box icon (3). You will then see a dialogue that shows all the fields in your selected record type (Figure 17)

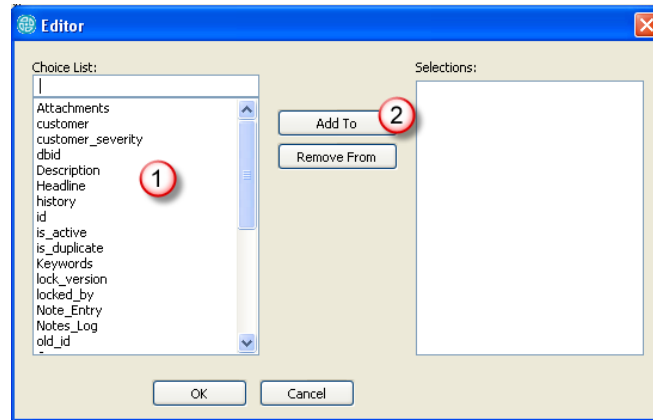


Figure 17. Field selection choice list

Select a field you're interested in ¹ and then click the Add To button ² to add it to the selected list of fields. Repeat these steps for the other fields you are interested in. Finally, click OK to return to the previous screen. The chosen fields will be displayed in the Fields Changed pane ³. <In the future, avoid explaining what your readers can be expected to know. Documenting "Add to" and "Remove from" is as superfluous as explaining how to use "OK" and "Cancel." As well as superfluous instructions, this document has a lot of verbiage that conveys no information; e.g., we see "It is recommended that you use.." where we should be seeing "Use...">

4.4 Defining Rule Phrases Using Actions and States

The EmailPlus Rule – Actions / States tab allows you to check for specific changes in state or for particular actions being performed. Figure 18 shows an example of this tab.

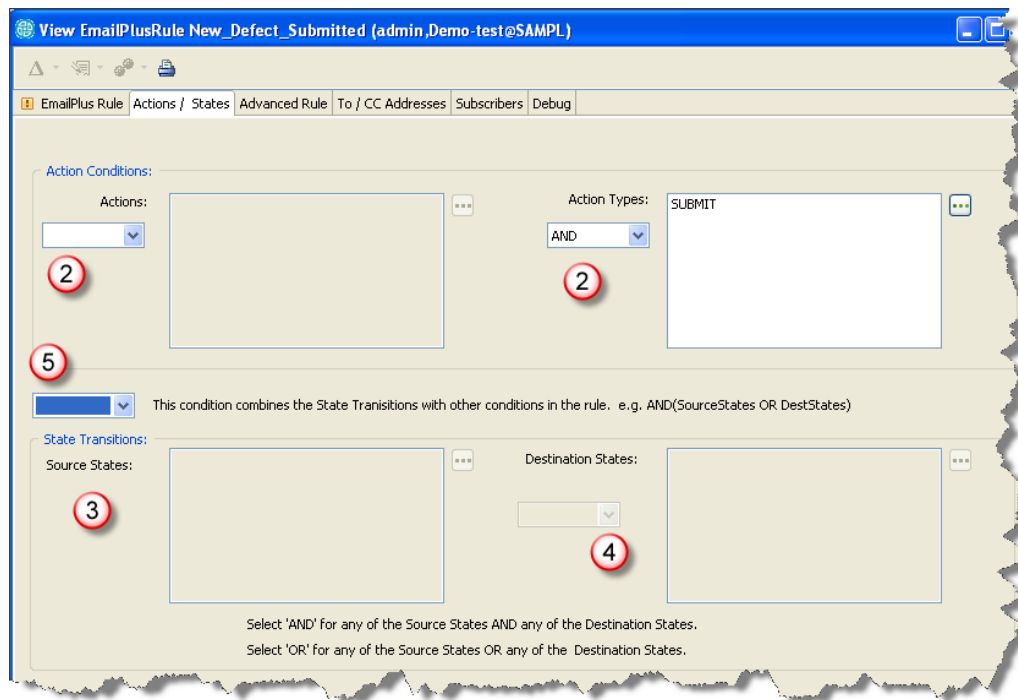


Figure 18. EmailPlus Rule Record - Actions / States Tab

4.4.1 Action Conditions

Two kinds of action conditions can be specified:

Actions: ①

If the record was updated using one of the action names (for example: Submit, Close, Open, Assign) specified in the list then the rule phrase will evaluate to TRUE.

Action Types: ②

If the record was updated using one of the action types (for example: CHANGE_STATE, MODIFY) specified in the list then the rule phrase will evaluate to TRUE.

As previously, in each of the cases above the phrase operand (AND/OR) must be selected first in order to make the rule phrase editable. You will then be able to choose the list of actions / action types associated with the rule phrase by clicking the multi-select icon. This will display a list of the possible choices.

4.4.2 State Transitions

The State Transitions section of the Actions / States tab allows the option of choosing to add a rule phrase based upon the source and / or destination states associated with the record being updated. To make the state transitions section editable the required AND/OR phrase operand ⑤ needs to be selected.

The state transitions rule phrase has the following format:

(<Source States> <AND/OR operand> <Destination States>)

The phrase is enclosed in brackets () to indicate that the whole phrase is evaluated as TRUE or FALSE. The rule can include either source states or destination states, or both:

<Source States> ③

Specify any source states which are needed to fire the rule. If no source states are required the list can be left blank.

<AND/OR Operand> ④

If the rule needs to include destination states then ensure the required AND/OR operand is selected.

AND

In order for the complete state transitions rule phrase to evaluate to TRUE **both** the source state for the record and the resultant destination state must appear in the respective lists of source and destination states.

OR

In order for the complete state transitions rule phrase to evaluate to TRUE **either** the source state **or** the resultant destination state must appear in their respective lists.

<Destination States> ④

Specify any destination states which are needed to fire the rule. If no destination states are required the list can be left blank.

4.5 To/CC Addresses Tab

The To / CC Addresses tab (Figure 19) allows you to specify who should receive email notifications when the rule is evaluated and found to be TRUE.

Figure 19. EmailPlus Rule Record - To / CC Addresses Tab

There are a number of ways that you can elect to notify users. Many of these are in common with the out-of-the-box ClearQuest Email package.

Fields:


Clicking the multi-select icon next to the Fields pane (see 1 in Figure 19) allows you to select fields from the ClearQuest record that are of type *Reference to Users*.

EmailPlus also searches referenced records for fields of type *Reference to Users*. So, for example, if you had a reference to a project record and the project record contained a field called ProjectMembers which was of type *Reference to Users*, then this field would also appear as an option Project.ProjectMembers. This facility is configurable. See Table 10 - EmailPlus Features Supported by Mail Transport Types.

When the EmailPlus rule evaluates to TRUE, the ClearQuest user identified in the selected field(s) will receive a copy of the email notification as a primary recipient.

Just to the right in the CC Addressees column a similar action will send a notification to the user as a CC addressee.


Groups:

Clicking the multi-select icon next to the Group pane  allows you to select one or more ClearQuest groups.


In this case, when the EmailPlus rule evaluates to TRUE, everyone in each of the selected groups will receive an email notification as a primary recipient.

Again, to the right in the CC Addressees column, a similar action will result in users being notified as a CC addressee.

Internet:


Clicking the multi-select icon next to the Internet pane  allows you to enter arbitrary Internet style addressees. You would typically use these for recipients who aren't ClearQuest users or who do not have a ClearQuest login of their own. Again, the same thing applies in the CC Addressees Column.

Project Role:

Finally, clicking the multi-select icon next to the Project Role pane  will allow you to select the project roles whose members will be notified when the EmailPlus rule fires.

This facility is only available if the companion ClearQuest package ProjectRoles is also installed in your schema. Otherwise these fields will remain read-only. The project name is taken from the record just edited and this is used to find the role membership for the required project.

EmailPlus does not provide a way for arbitrary ClearQuest User Ids to be added to the addressees since this facility is provided by the self subscription mechanism described in section 4.6. Subscriptions can be defined on either to the EmailPlus rule record or the record types to which the EmailPlus package has been applied. This reduces administrative load since this is a self-service operation.

It is also possible to generate a CC Copy for the person who performed the action on the record that caused the notification to be generated by ticking the CC Actioner checkbox .

Note: In the event that one address appears in both the To and CC lists, the CC copy will be removed. This should ensure that any given person only gets one copy of any notification.

4.6 Subscription

The Subscribers tab (Figure 20) gives the EmailPlus administrator control over the subscription facility. In order for it to be possible for individuals to be able to subscribe to a rule, the administrator must create a subscription record for each site (or replica) where it is required that the facility be available. If no subscription record(s) exists, then the subscription facility won't be available for that rule.

Firstly the administrator must create the subscription record. This is done by clicking the *New* button **1**.

The administrator can also restrict who is allowed to subscribe to a rule by specifying that prospective subscribers must belong to one of a list of specified ClearQuest user groups **2**. This is done so that users, who are not normally able to see a record because of ClearQuest security contexts, can't subscribe to a rule and circumvent this.

Note: This mechanism only prevents a person subscribing; it will not automatically remove them if the group memberships subsequently change. This remains a manual operation.

If no subscription groups are specified, then anyone can subscribe.

If you have created a *Special Interest* EmailPlus template for an enabled record type, users will be able to personally subscribe to be notified about changes to individual records. They do this by selecting the particular record(s) and adding themselves to the records subscription list. The user must be able to perform some action on the record to be able to do that. If you haven't created a *Special Interest* template then this type of subscription will be refused.

Note: If you remove a Subscriber list entry, the referenced subscription record is also deleted if possible. In ClearQuest MultiSite environments it is not possible to delete subscription lists that are mastered elsewhere. In this case, removed subscription records must be manually deleted before it will be possible to create a new subscription record for that remote site. You cannot reattach a subscription list record to a rule once it has been removed.

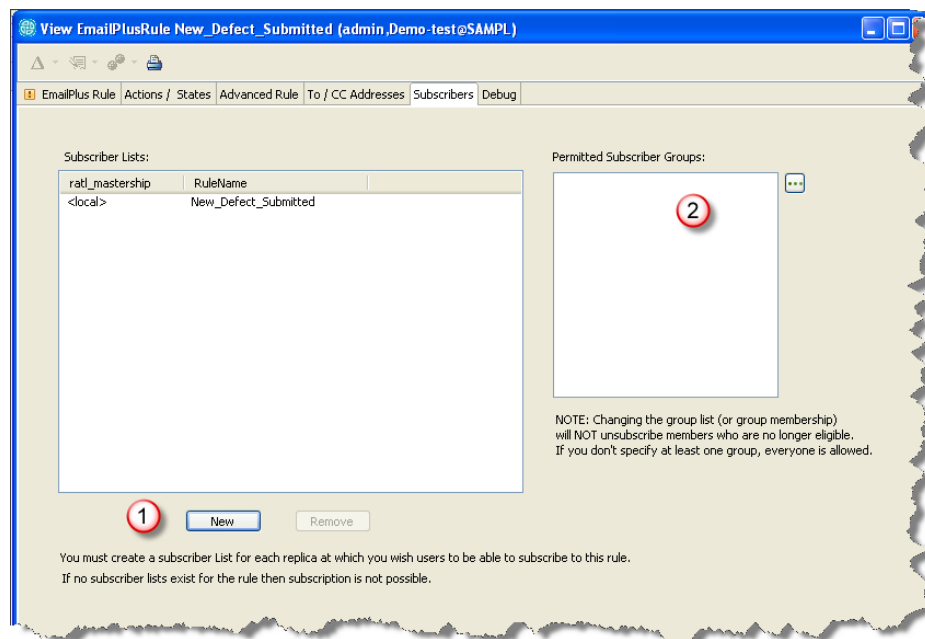


Figure 20. EmailPlus Rule Record - Subscribers Tab

4.7 Defining a Rule Phrase Using an Advanced Rule

The Advanced Rule tab (Figure 21) provides the final method for defining an EmailPlus rule phase. This may seem to be a very simple form, but it is at the very heart of EmailPlus rules. This is really what gives it its power. <Just the bare facts, please; characterizing a product or feature can open the door to litigation. Leave it to Marketing Communications to characterize products—they know how to avoid the legal pitfalls.>

View EmailPlusRule New_Defect_Submitted (admin,Demo-test@SAMPL)

EmailPlus Rule Actions / States Advanced Rule To / CC Addresses Subscribers Debug

1 Select appropriate boolean operator if you want to use an Advanced Rule component.
The rule is added to the rest of the expression as either: "AND (Advanced Rule)" or "OR(Advanced Rule)".

*Advanced Rule:

2

Validate Rule This button allows you to validate the Advanced Rule expression.

3 VALIDATED

The Advanced Rule can be *ANY* valid Perl Boolean expression. You can use *ANY* Perl functions and *ANY* functions in your schema's global scripts. This allows you to create very rich conditional expressions.

Figure 21. EmailPlus Rule Record - Advanced Rule Tab

The advanced rule allows you to combine *any* valid PERL expression with the rest of the rule conditions. In fact all of the other parts of the rule could be expressed as a PERL expression. They are there more for your convenience. This means you can construct *any* expression you like and that expression can:

- Call *any* built-in PERL function,
- Call *any* function defined in the global scripts section of your schema, and
- Call *any* function from the PERL ClearQuest API itself, or any other object available to PERL.

Additionally, the expression created using the advanced rule is not limited to working with just the record being edited.

To use the Advanced Rule select the required AND/OR phase operand 1 to combine it with the rest of the rule and then type your expression in the space provided 2.

You can check whether the expression will work by clicking on the Validate Rule button provided 3. If everything is OK, then the string **VALIDATED** will appear next to the button else the PERL expression evaluation error will be displayed.

To make things a bit easier for you, a number of useful functions have been added to the global scripts of the EmailPlus package (Table 2).

Function	Description
OneOf(\$list, \$item)	<p>This routine takes a list and searches it for the presence of the given item. It returns TRUE if the item is found else FALSE.</p> <p>\$list - A reference to an array of string values</p> <p>\$item - The value to search the list for</p>
Gfv(\$fieldName)	<p>Simple encapsulation of GetFieldValue which is equivalent to:</p> <p>GetFieldValue(\$fieldName)->GetValue()</p> <p>\$fieldName – The name of the field</p>
Gfov(\$fieldName)	<p>This is similar to Gfv, except it is the simple encapsulation of GetFieldOriginalValue which returns the original value of a field before any changes were made.</p>
Gfvs(\$fieldName)	<p>This function is the simple encapsulation of ClearQuest API call GetFieldValueStatus and is equivalent to:</p> <p>GetFieldValue(\$fieldName)->GetValueStatus()</p>
FChg(\$list)	<p>Examines the list of field names given and returns a string to indicate what fields have changed:</p> <p>"ANY" means one or more of the fields in the list have changed</p> <p>"ALL" means all the fields in the list have changed</p> <p>"NONE" means none of the fields have changed</p> <p>\$list - A reference to an array of field names to check for change.</p>
StoDT(\$dateString)	<p>Converts a date string value from a ClearQuest DATE_TIME field to a UTC date/time in seconds. The return value is the number of non-leap seconds since the epoch. On most systems the epoch is 00:00:00 UTC, January 1, 1970.</p> <p>\$dateString - A date string of the format "YYYY-MM-</p>

	DD hh:mm:ss”.
DTtoS(\$timestamp)	Converts a UTC Date/Time in seconds into a date string of format YYYY-MM-DD hh:mm:ss. \$timestamp – A UTC date/time in seconds.

Table 2. EmailPlus Rule Functions

5 The Administrator Group

All administration activities for the package are controlled by membership of the ClearQuest user group called **EmailPlusAdmins**. This group should be created and then subscribed to all databases that use a schema to which the EmailPlus package has been applied. This separate group is employed so that administration of EmailPlus capability may be delegated to specific individuals. If you already have a ClearQuest administrators group of some kind and want the same group of people to administer EmailPlus, simply make your existing administrators group a member of the **EmailPlusAdmins** group.

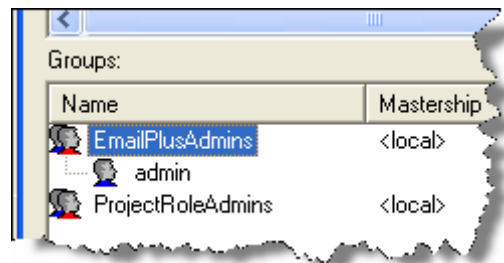


Figure 22. EmailPlusAdmins ClearQuest Group

6 Applying the EmailPlus Package

Since the package isn't part of the out-of-the-box ClearQuest installation, there are some manual steps you need to perform to make the package available to ClearQuest. These steps only need to be done on the machine you will be using to add the package to your schema. Once the package is applied, it is copied into your schema and hence available to all other clients. However, please be aware that any third party components used by the package will need to be installed separately.

Note: If you ever export your schema with the intent of creating a new schema from it in a new schema repository, you will also need to provide a copy of the package install files for the recipient to be able to import your schema.

6.1 Installing the Package

The package is supplied as a zip file containing all the necessary files. To install the package, you should start by referring to the *IBM Rational ClearQuest EmailPlus Release Notes & Installation Guide* to unpack the zip file and save the files in the correct locations. Then proceed as follows:

1. Restart the designer if it was already running when you installed the package files.

2. Use ClearQuest Designer to register the new package by starting the package wizard (see Figure 23).

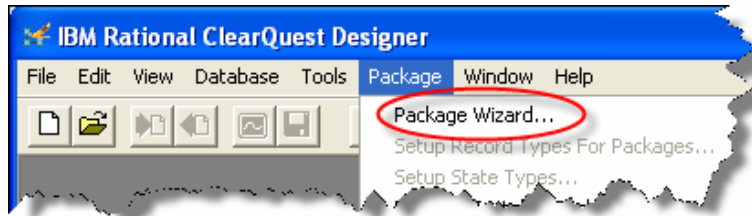


Figure 23. ClearQuest Designer Start Package Wizard

3. In the package wizard, if the EmailPlus package is not listed select More Packages... (see Figure 24).

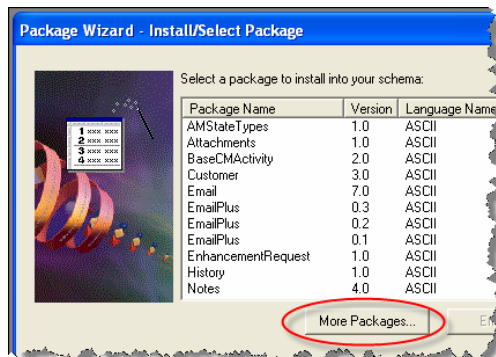


Figure 24. Locate Unregistered Packages

4. Find the newly installed package, select it and click OK (see Figure 25).

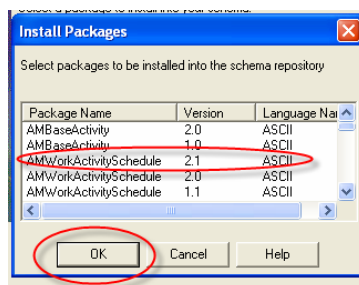


Figure 25. Selecting the package

Once the package is installed and registered it can now be applied to the record types in the schema.

6.2 Applying the Package to your schema

To apply the package to your schema, perform the following steps:

NOTE: In a ClearQuest MultiSite environment, the package should only be registered at the working master site.

1. Start the ClearQuest designer and then start the package wizard (see Figure 26).

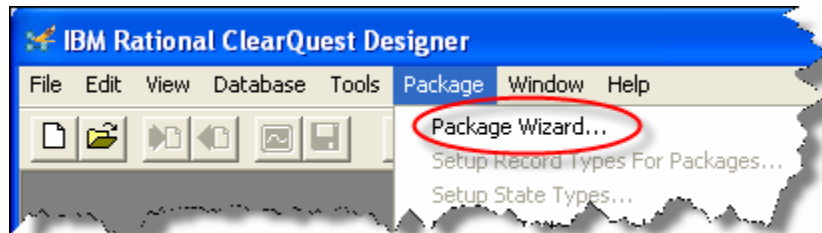


Figure 26. Starting The Package Wizard

2. Select the EmailPlus package to be applied and click 'Next' (see Figure 27).

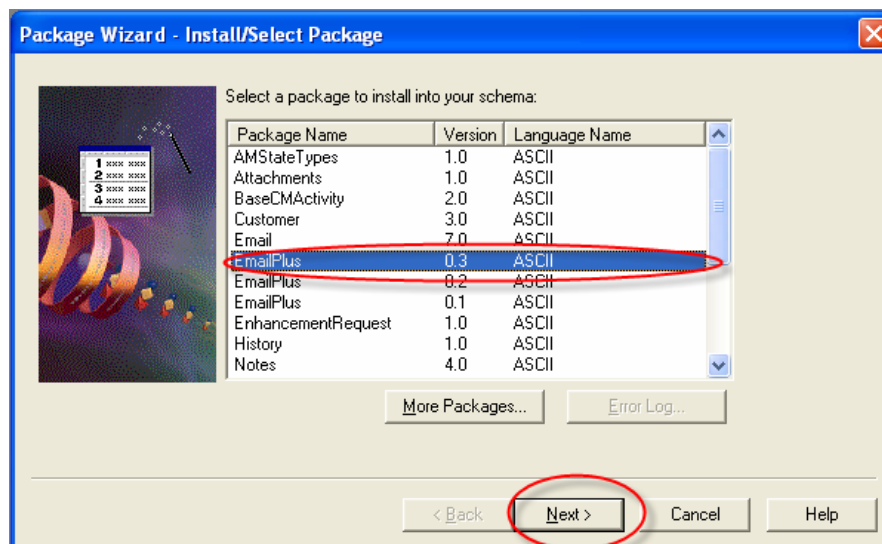


Figure 27. Package Wizard - Selecting The EmailPlus Package

3. Select the schema version you want to apply the EmailPlus package to and click NEXT (see Figure 28).

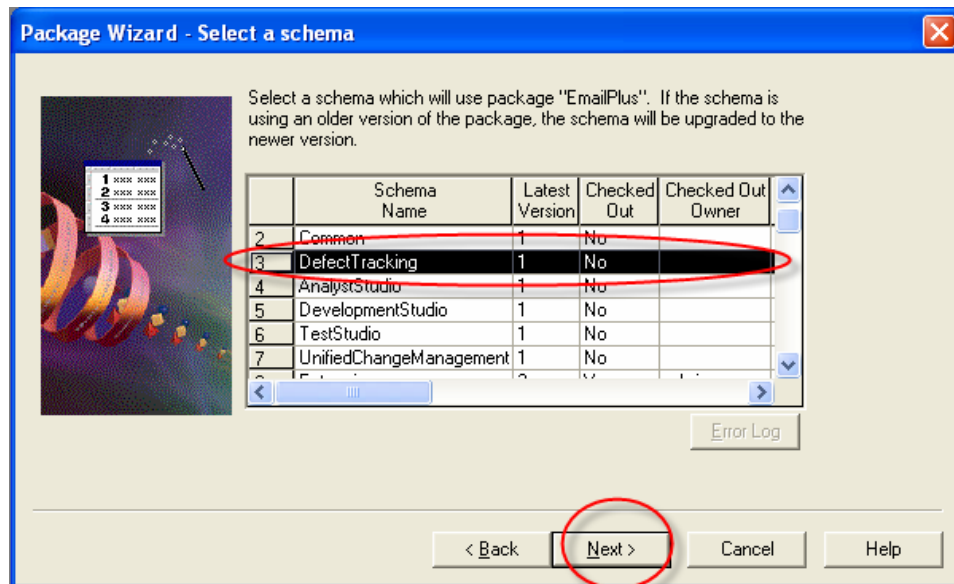


Figure 28. Package Wizard - Selecting the schema which will use EmailPlus

4. Now enable all the record types that you want the package to work with (see Figure 29). Click FINISH to complete the process.

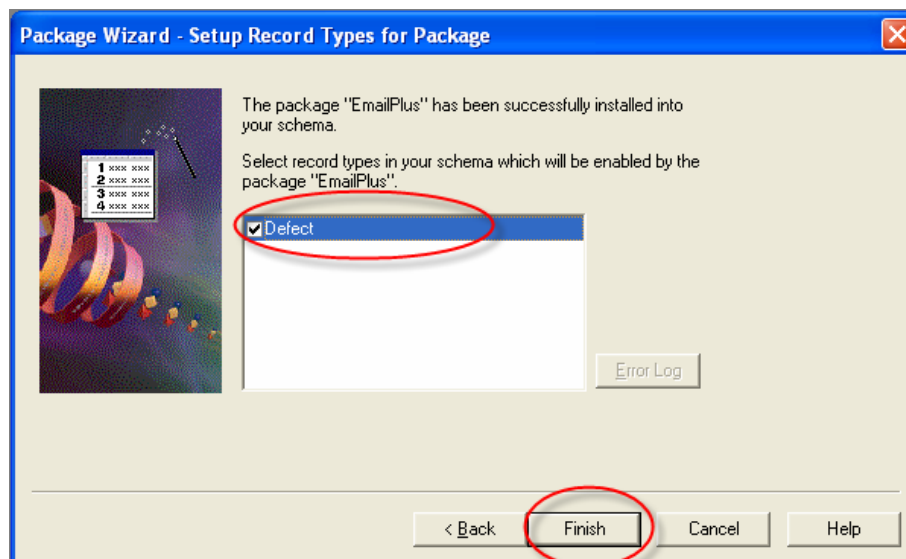


Figure 29. Package Wizard – Enabling record types with EmailPlus

6.3 Enabling a Stateless Record Type

The EmailPlus package can be applied to both stateful and stateless record types. For stateless record types, the package needs to be applied manually to the record type using the steps described in this section.

Note: It is only meaningful to enable stateless based record types in a schema whose script language is set to PERL. This won't work in a VBScript based schema because as these have to be added manually, they aren't able to take advantage of the independence

associated with the package's scripting language. Moreover, this script cannot be converted to VBScript since VBScript cannot call the PERL routines in the package.

To enable stateless record types manually, perform the following steps:

1. Create a new multiline string field – you can call it what you like, the package would have called it ***SubscriberList***.
2. Create a record script called 'EMP_AddMe' and put the following code in it.

```
# Add the current user to the list of those requesting notification
$result = EMP_Special_AddMe("SubscriberList");
```

Substitute your field name for "SubscriberList" if you called it something else.

3. Create a record script called 'EMP_RemoveMe' and put the following code in it.

```
# Remove the current user from the list of those requesting
# notification
$result = EMP_RemoveMe("SubscriberList");
```

Substitute your field name for "SubscriberList" if you called it something else.

4. Add the field 'SubscriberList' to a tab on your form(s) and make it read-only for all users. Use a ListBox control.
5. Finally add a new base action called EmailPlusNotify and create a Notification hook with the following code:

```
# This hook determines if the criteria for email generation are met and
# then assuming that they are, generates an email to all those people
# on the auto notify list
#
# The first parameter is an array of fields of type reference to user.
# If the value of any of these fields is changed, then a change of
# ownership email will be sent to the new and old owners
# The second parameter is the name of the notification list field.
# This field contains a list of email addresses of people who have
# subscribed to be notified about any change to the record
EMP_ExecuteNotification(undef , "SubscriberList");
```

Note: If you called your field something other than SubscriberList then you should substitute the name you used in each of the places used in the code fragments above.

Note: You must still have applied the EmailPlus package to your schema.

6.4 Configuring an Enabled Record Type

There is a small amount of customization required to any record type enabled by the package.

1. If you don't want users to have the ability to subscribe to notifications associated with an individual record, either hide or restrict access to the Subscriber List tab. If you leave the tab visible and don't create a Special Interest template, user subscription will be refused.
2. If you want to update a specific record type to enable owner notifications, locate the following line of code in the EmailPlusNotify Notification hook:

```
EMP_ExecuteNotification(undef, "SubscriberList");
```

Replace the first argument “undef” that is passed to the EMP_ExecuteNotification function with a reference to an array which includes all the fields you regard as ownership fields for the record type. For example,

```
EMP_ExecuteNotification( [ "Owner", "Submitter" ], "SubscriberList");
```

3. Provided an ownership type EmailPlusTemplate record exists for the enabled record type, the package will generate notifications whenever any of the listed fields change. The notifications are sent to the old and new owners.

Note: An email will be generated even if originally one of the designated owner fields was blank or becomes blank. Any change to any of the designated owner fields will trigger the email generation if an ownership template exists for the enabled record type.

6.5 Recommended Testing Strategy

Because ClearQuest packages cannot be removed from your schema once applied (except by deleting the schema version they were applied to and all later versions), it is desirable that you should ensure the package does what you want and will work in your environment *before* you apply it to your live schema.

To this end, one of these strategies is recommended for evaluating the package before applying it to your live schema.

It is a prudent first step to take a full backup of ALL your ClearQuest databases before you apply the package.

6.5.1 Copy Live Environment

This strategy involves making an entirely independent and complete copy of your live environment databases so that the effects of the package can be judged. To do this, make copies of your schema repository and user databases using the command line utility ‘installutil’. The process for doing this is documented in [Tech Note #1118690](#).

Note: If you are in a MultiSite environment, you should ensure that there is *no possibility* of shipping updates packets from your copy to live environments.

Please be aware that copies of ClearQuest databases are dangerous things to use until you have updated the database credentials in the copy of the schema repository. Until these are updated, the copy of the schema repository points at your LIVE databases and they are at risk if they can be accessed.

If you don't feel happy with this process seek advice or use the Copy Schema strategy outlined below.

6.5.2 Copy Schema

This strategy is somewhat simpler than the copy live environment strategy, but requires potentially more work to establish the environment.

1. Using the ClearQuest Designer, create a new schema based upon the most recent version of your live schema.
2. Apply the package to the newly created copy of your schema.

3. Create a new database and base it on the newly created copy of your schema.
4. Try the amended schema out in this new database and verify it will work in your environment.
5. If everything checks out, you can apply the package to your live schema and start benefiting from EmailPlus.

7 Mail Relay

In some IT environments access to Simple Mail Transfer Protocol (SMTP) servers is strictly controlled. This often means that the ClearQuest clients may not directly access the SMTP server in order for email notifications to be sent without the IP address of the client machine being registered with the SMTP server or a firewall surrounding it.

In these kinds of environments, IT departments are often unwilling to register every ClearQuest client machine, since; the whole purpose of this restriction is to reduce the risk of the SMTP server being used to send SPAM.

In these instances the only options are to:

- Implement your own SMTP server for ClearQuest emails. This is likely to be frowned upon for this just creates a new target for SPAM
- Implement some kind of private relay service to forward the email request to a server that is registered for SMTP server access. This cannot simply be an SMTP relay since otherwise the same problems would exist.

The EmailPlus package implements the second of these options in two different ways. These are implemented using two different transport types: **SMTP Relay** and **PostOffice Relay**.

7.1 SMTP Relay

This transport type is implemented using a Windows application. For now, this application must be started manually in order for it to receive email relay requests.

When this transport method is used, the EmailPlusSiteConfig fields for the SMTP server are used in a slightly different way. The SMTPServerName field now becomes the name of the server where the relay service is installed. Similarly, the SMTPPortNumber field is the port number the mail relay service will listen on. It is important that the mail relay service is configured to use the same port as defined in your EmailPlusSiteConfig record(s).

Figure 30 shows the communications path taken to deliver mail.

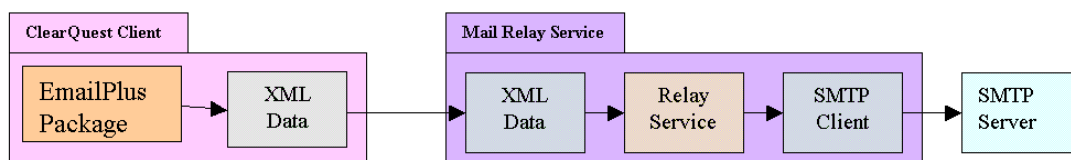


Figure 30. SMTP Relay Architecture

When the ClearQuest client executes the hook code in the EmailPlus package, it calls upon the installed component to establish a link with the Mail Relay service that receives the email encoded in an XML stream. This is then decoded by the relay service and email sent to the SMTP server using either the SMTP client provided by the ClearQuest mail transport or one provided in the installed component.

The default port number used is 36001. Whatever port you select, that is the only one required for incoming requests should the relay service be behind a firewall. Obviously, access to the SMTP server port configured will be required for outgoing requests.

7.2 PostOffice Relay

This transport type seeks to achieve the same kind of results as the SMTP Relay transport type except that it does it in a completely different way.

The prime reasons for this transport type are to:

- Increase resilience of generated Emails. Since they are registered in the ClearQuest database, they cannot be lost even if the relay service goes offline for a while. It will simply pick up where it left off and deliver all the outstanding emails once it is restarted.
- Eliminate the need for an installable component from client desktops. You still have the choice of whether or not to use the installable component or CQMail transports in the relay component itself. Provided you don't mind losing the HTML body part and embedded graphics / attachments, you can eliminate the need for the installable component entirely.

Figure 31 shows the communications path taken to deliver mail.

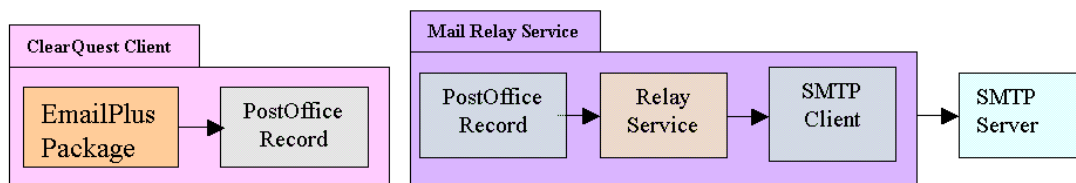


Figure 31. PostOffice Relay Architecture

7.3 EMP Relay Application

The EMP Relay application is a Windows-only application that is responsible for receiving XML encoded mail notifications from your EmailPlus enabled ClearQuest database and then translating those requests into actual emails. This application is intended to be used in environments where access to standard SMTP servers may be restricted. The EMP Relay application is intended to be used with the following EmailPlus mail transport modes:

- **Network Relay Mode**
The service listens for incoming network connections from the EmailPlus enabled ClearQuest database. It accepts the XML encoded information, then decodes it and establishes a connection to the SMTP server to send the mail.

- **PostOffice Relay Mode**

The service periodically polls your EmailPlus enabled ClearQuest database for EmailPlusPostOffice records. Each EmailPlusPostOffice record contains an XML encoded mail notification. The data contained in this record is decoded and then forwarded via email to the SMTP server.

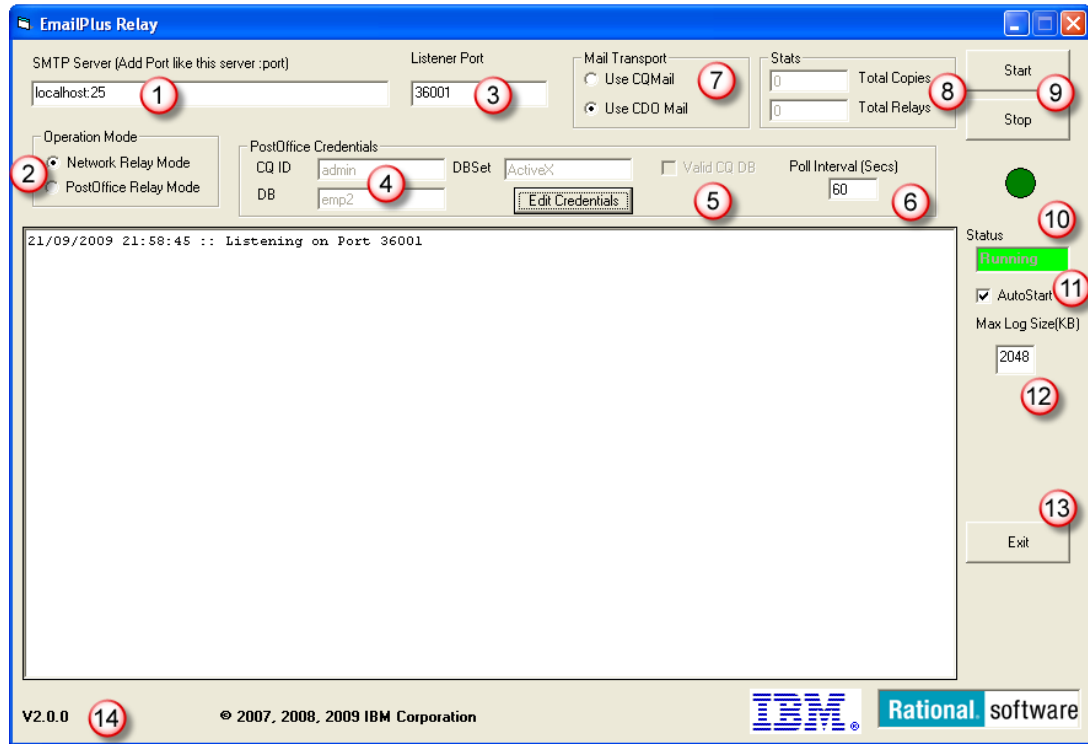


Figure 32. EmailPlus Relay Interface

Table 3 explains the EmailPlus Relay Interface shown in Figure 32.

Field	Definition
①	Specifies the name of the SMTP server and port to use. This is only used by the CDO³ Mail transport option. If using the CQMail transport then the ClearQuest client settings are used.
②	This area selects the Operating Mode of the tool. <ul style="list-style-type: none"> • When in Network Relay Mode the service listens for incoming forwarding requests on the port specified ③. • When in PostOffice Relay mode, the service checks the designated ClearQuest database for PostOffice requests.
③	This is the port the service listens to for incoming forwarding requests. This is only used in Network Relay Mode . This must match the port given in the EmailPlusSiteConfig record.

³ CDO – Collaboration Data Objects – A Microsoft® Windows® component providing SMTP Client services.

4	<p>This area identifies the ClearQuest database containing EmailPlusPostOffice records to process. The database must be based on a schema that has had the EmailPlus package applied to it.</p> <p>This is only used in the PostOffice Relay mode. In order to edit the credentials click on the button.</p> <p>Note: The CQ Login ID provided must be a member of the EmailPlusAdmins group for the service to function correctly.</p>
5	<p>This indicates if the ClearQuest database credentials supplied are for a database that the service can run with. Only applicable in PostOffice Relay mode.</p>
6	<p>Specifies the database poll interval in seconds. This is only used in PostOffice Relay mode and determines the period between queries looking for post office records to process.</p>
7	<p>Selects the mail forwarding transport mode. In CQMail Mode, the emails are sent via the ClearQuest mail component that must have been correctly configured using the ClearQuest Client. In CDO Mail transport mode the mail is forwarded to the SMTP server identified in 1. This transport mode affords the greatest level of functionality.</p>
8	<p>Provides statistics surrounding the amount of mail sent. The statistics are only reset when exiting the application.</p> <ul style="list-style-type: none"> • Total Copies records the total number of mail recipients of all emails sent. • Total Relays records the number of mail requests.
9	<p>The Start and Stop buttons start or stop all forwarding. Pressing the Start button causes an immediate database poll when in PostOffice Relay mode.</p>
10	<p>These are indicators of the current status of the service.</p> <ul style="list-style-type: none"> • The green circle flashes once a second to indicate that the service is alive. • The status box indicates if the service is forwarding mail or not. <ul style="list-style-type: none"> ○ When the status box contains the word running the service is actively forwarding emails. ○ When the status box is highlighted red it isn't.
11	<p>The auto start tick box indicates whether or not the service should enter the running mode immediately without any intervention using the settings that were in place when it was last run.</p>
12	<p>This is the size in KB of the space allocated to the log. The minimum size is 16 KB but this figure can be increased if desired. There is no facility to save the log although you can select the content and use cut and paste if required.</p>
13	<p>The Exit button causes the application to terminate. Current settings are saved at this point. The application may not terminate immediately if it is actively</p>

	processing requests.
14	This is the version number of the application. Please include this in any problem reports.

Table 3. EmailPlus Relay Interface Window Definitions

8 Dependencies

The EmailPlus package is optionally dependent on another ClearQuest package called ProjectRoles⁴ <Do not mention unreleased software>. This package allows the administrator to define a set of roles for use with the schema and permits mappings of actions to role for each project. This dependency appears in the EmailPlusRule record type on the To/CC Address tab. If the package is installed, EmailPlus automatically detects this and makes available an option to send emails to the membership of a project role. This capability is more fully described in the ProjectRoles⁷ package documentation <which customers can't read!> and in section 4.5.

9 Performance

This EmailPlus package is implemented making extensive use of caching techniques to avoid continually retrieving information from the ClearQuest database and thus minimize the impact of evaluation and execution of the email rules on the ClearQuest environment.

However, be aware that since the email rules provide significant flexibility, care should be taken when constructing the rules to ensure that any *Advanced Rules* minimize their own impact on performance.

EmailPlus caches all rules for any given record type in memory the first time they are examined in any ClearQuest session. No matter how many rules there are for a single record type, only 1 record is fetched from the database.

The Simple EmailPlus rule phrases are not evaluated using ClearQuest queries. Evaluation of these phrases is undertaken entirely in memory.

If you make use of the PostOffice transport mode you should consider the following:

- There is an additional load on the database. Each email generated results in a record (and its history) being created in the ClearQuest database. There is a corresponding cost when the relay service processes the record and subsequently deletes it. Additionally, every time the relay service polls the database for EmailPlusPostOffice records, a ClearQuest query is run against the database. The administrator needs to find a balance between the number of queries and the number of records processed at one time to even out load. The additional load presented to the database will be dependent on the number, kind and frequency of email rules generating messages. This balance will have to be achieved by a process of trial and error. Ideally the relay service should not be processing large numbers of records in a batch, nor should it be running excessively frequently.

⁴ The package ProjectRoles has not yet been released. This is a placeholder for future expansion.

- If Post Office records are not processed by a local instance of the relay service, then emails will be delayed by approximately one MultiSite replication cycle period.

10 Email Attachments

EmailPlus 2.0 includes the ability to send ClearQuest record attachments in emails when using one of the following email transport types:

SMTP Direct

SMTP Relay

PostOffice

The EmailPlus Template records also allow you to include attachments, such as images when generating HTML emails with inline images. The record and template attachments are cached to the ClearQuest client when an EmailPlus notification is sent. Due to this feature there are a number of things to consider when enabling attachments in EmailPlus.

Security:

Due to the confidential nature of some record attachments you will need to take precautions to ensure that record attachments are not sent to people who do not have authority to view them.

Performance:

When attaching record attachments to EmailPlus notifications you should consider the performance of record modification due to the fact that the ClearQuest client will download the record attachments to your local machine or ClearQuest Web server when constructing the EmailPlus notification. If attaching large record attachments to EmailPlus notifications then the performance of downloading the attachments to the ClearQuest client will be impacted by network speeds and proximity to the ClearQuest server. You should consider how your users access ClearQuest in your environment.

10.1 Default ClearQuest Client Temporary Cache Directory

When attachments are included in an email the files will be downloaded to a temporary directory on the ClearQuest Client local machine. Table 4 and Table 5 illustrate the order of the defined default locations for these temporary directories.

Windows Location	Example
1. The value of the %TEMP% environment variable	C:\Documents and Settings\user name\Local Settings\Temp
2. The value of the %TMP% environment variable	C:\Windows\Temp

Table 4. Windows Temporary Cache Directory Location

UNIX/Linux Location	Example
1. The default tmp directory	e.g. /tmp/ClearQuest/EmailPlus/Cache

Table 5. UNIX/Linux Temporary Cache Directory Location

10.2 Specifying the ClearQuest Client Temporary Cache Directory

<Please notice how the following suggested revision eliminates both verbiage that conveys no information and the superfluous information (admins already understand rwx).>

To specify the location of the temporary cache directory, set the environment variable EMP_TEMPDIR to an absolute pathname.

Note: Use EMP_TEMPDIR on ClearQuest Web servers if you plan to send attachments in EmailPlus notifications.

10.3 ClearQuest Client Temporary Cache Directory Housekeeping

Many of the record attachments will be deleted after they have been included in an EmailPlus notification. However, it should be noted that template attachments are cached for the duration of a ClearQuest session to improve performance and may not get deleted. As such it is good practice to periodically remove any of the files from the temporary directory to prevent any filesystem capacity issues. This housekeeping could be done using a cron job on Linux/Unix or a scheduled task on Windows.

Caution should be exercised when deleting these files as some of the files may still be needed by open ClearQuest sessions, so as a recommendation you should only delete the cached attachment files if they are over 24 hours old.

Appendix A – Template Meta Tag Substitution Extension Point

```
sub EMP_SubstituteCustomMetaTag {
    my $metaTagParts_ref = shift; # a reference to an array
    my $entityDefObj      = shift;
    my $isHTMLContent     = shift;
    my $isTest            = shift;

    # This routine provides a custom extension to the Meta Tag substitution processing.
    # In the event that none of the built in meta tags is sufficient for the CQ Administrator's needs
    # this extension allows administrators to add their own meta tags. You should explore the
    # EXPRESSION meta tag beforehand though since it is envisaged that most needs will
    # be met using that facility. Bear in mind that you can call functions defined in the
    # schema as well. It is envisaged that this customisation point would be used where it
    # is desired to be able to test the function and / or switch between HTML output and plain text.
    #
    # Parameters:
    # metaTagParts_ref - This is a reference to an array of the decoded parts of the Meta tag as read from the
    #                   template provided. The 0 element is the name of the meta tag whilst any other
    #                   elements that may be present hold parameters decoded from the original input.
    #                   The original input is split on "::" boundaries. The complete original tag can
    #                   be reconstituted with this statement if desired
    #                   Orig = Join(MetaTagParts, "::")
    # entityDefObj      - This is the ClearQuest Entity Def object for the record type to which the template pertains
    # isHTMLContent     - This is a boolean flag that when set to 1 means you can render the result as properly formed
    #                   HTML if desired. Otherwise whatever you return will be treated as plain text.
    # isTest            - This is a boolean flag and when set to 1 indicates that you should return any Error that the expanding the Tag may
    #                   generate AND NOT the value. Returning a null string in isTest mode means there were no errors
    #                   This feature is used to validate the usage of a tag.

    my $errorTags = ""; # Put any error text in this variable
    my $fieldValue = ""; # Put the result of the tag expansion in this variable

    # Process the custom tags

    my $customTag = lc($metaTagParts_ref->[0]);

    # Add your processing here
    # =====
    # Note that the Meta Tag is converted to lower case so all of your literal values should also be in lower case
    # this handles the situation when the template rule writer uses mixed or upper case for the tag name

    # =====
    # This dummy is for test only and can be removed START->

    if ($customTag eq "dummy") {
```

```
    $fieldValue = "Customised test tag ".localtime();
  }
  elseif ($customTag eq "dummy2") {
    $fieldValue = "Customised test tag ".localtime();
  }
  # <- END This dummy is for test only and can be removed
  # =====
  else {
    # Catches Undefined tags
    $errorTags .= "UNKNOWN Meta Tag Substitution ".$metaTagParts_ref->[0]."\n";
  }

  # Return either value or error string dependent on mode
  my $result = ""; # The return value
  if ($isTest) {
    $result = $errorTags;
  }
  else {
    $result = $fieldValue;
  }
  return $result;
}
```

Appendix B – Regular Expressions Meta Characters

Table 6 details the some of the standard escape characters that can be used in PERL Regular Expressions. These are all supported in the Email-PlusRule and EmailPlusTemplate records.

Meta Character	Description
\	Marks the next character as either a special character or a literal. For example, "n" matches the character "n". "\n" matches a newline character. The sequence "\\" matches "\" and "\(matches "(".
^	Matches the beginning of input.
\$	Matches the end of input.
*	Matches the preceding character zero or more times. For example, "zo*" matches either "z" or "zoo".
+	Matches the preceding character one or more times. For example, "zo+" matches "zoo" but not "z".
?	Matches the preceding character zero or one time. For example, "a?ve?" matches the "ve" in "never".
.	Matches any single character except a newline character
(pattern)	Matches pattern and remembers the match. The matched substring can be retrieved from the resulting Matches collection, using Item [0]...[n]. To match parentheses characters (), use "\(" or "\)".
x y	Matches either x or y. For example, "z wood" matches "z" or "wood". "(z w)oo" matches "zoo" or "wood".
{n}	n is a nonnegative integer. Matches exactly n times. For example, "o{2}" does not match the "o" in "Bob," but matches the first two o's in "foooooo".
{n,}	n is a nonnegative integer. Matches at least n times. For example, "o{2,}" does not match the "o" in "Bob" and matches all the o's in "foooooo." "o{1,}" is equivalent to "o+". "o{0,}" is equivalent to "o*".
{n,m}	m and n are nonnegative integers. Matches at least n and at most m times. For example, "o{1,3}" matches the first three o's in "foooooo." "o{0,1}" is equivalent to "o?".
[xyz]	A character set. Matches any one of the enclosed characters. For example, "[abc]" matches the "a" in "plain".
[^xyz]	A negative character set. Matches any character not enclosed. For example, "[^abc]" matches the "p" in "plain".

[a-z]	A range of characters. Matches any character in the specified range. For example, "[a-z]" matches any lowercase alphabetic character in the range "a" through "z".
[^m-z]	A negative range characters. Matches any character not in the specified range. For example, "[m-z]" matches any character not in the range "m" through "z".
\A	Match only at beginning of a string
\b	Matches a word boundary, that is, the position between a word and a space. For example, "er\b" matches the "er" in "never" but not the "er" in "verb".
\B	Matches a non-word boundary. "ea*r\b" matches the "ear" in "never early".
\d	Matches a digit character.
\D	Matches a non-digit character.
\f	Matches a form-feed character.
\n	Matches a newline character.
\r	Matches a carriage return character.
\s	Matches any white space including space, tab, form-feed, etc.
\S	Matches any non white space character.
\t	Matches a tab character.
\v	Matches a vertical tab character.
\w	Matches any word character including underscore. Equivalent to "[A-Za-z0-9_]".
\W	Matches any non-word character. Equivalent to "[^A-Za-z0-9_]".
\z	Match only at the end of a string.
\Z	Match only at the end of a string, or before newline at the end.

Table 6. Perl Regular Expression Meta Characters

For more information regarding PERL Regular Expressions please review the PERL documentation (<http://perldoc.perl.org/>).

Appendix C – Features / Mail Transport Matrix

Table 7 shows the features available for each of the transport types. The custom variant is blank since what it supports depends on your implementation.

	Mail Transport Types ⁵⁶					
Feature	CQMAIL	SMTP Direct	SMTP Relay ⁷	PostOffice Relay		Custom
				with CQMail	with CDO	
Templates	Y	Y	Y	Y	Y	
Email Rules Simple	Y	Y	Y	Y	Y	
Email Rules Advanced	Y	Y	Y	Y	Y	
HTML Content	N	Y	Y	N	Y	
Mail Headers	N	Y	Y	N	Y	
Embedded Graphics	N	Y	Y	N	Y	
Attachments	N	Y	Y	N	Y	
Mail Relay	Y/N ⁸	Y	Y	Y	Y	
Subscriptions	Y	Y	Y	Y	Y	
Multisite Support	Y	Y	Y	Y	Y	
UNIX Platform Support	N	N	N	N	N	N

⁵ In a ClearQuest MultiSite environment, you can have different transport types at different sites if you want. It will make it harder to manage and you may need to replicate rules that use different templates depending on the site the rule executes.

⁶ Please note that for all of the 'N' in this table, EmailPlus won't stop you from trying to use these features, it just won't work. HTML for example will come out as raw HTML, attachments and graphics will be ignored.

⁷ Assuming the Network Relay mode is enabled in the EMP Relay application, otherwise the CQMail restrictions will apply.

⁸ For mail relay, you can select CQMAIL as the delivery mechanism at the receiving end even though the transport type is set to SMTP Relay. If you do this you will lose some of the features by using the CQMAIL delivery. It is recommended that you choose SMTP Direct for the delivery mechanism.

Windows Platform Support	Y	Y	Y	Y	Y	Y
--------------------------	---	---	---	---	---	---

Table 7. EmailPlus Features Supported by Mail Transport Types

Appendix D – Configuring Addressing Options

The EmailPlus package allows you to address emails by identifying field(s) that are a reference or reference list to users. It also searches referenced record types for such references, but a configuration item limits the depth of recursion. As supplied EmailPlus searches

- The current record
- Its children
- Its grandchildren

If you wish to change this depth then you can define a constant in the PERL Global Script 'EMP_Customisable'.

```
$MAX_FIELDPATH_RECURSE_DEPTH = 2;
```

Setting the value to 2 (as shown in the example above) will set the depth of recursion to two, i.e. the child and grandchild records of the current record.

Setting the value to 0 will limit the search to the current record only.

Setting the value to 3 will search child, grandchild and great grandchild, in addition to the current record.

There is also another option to widen the kind of fields searched for:

```
$FIELDPATH_INCLUDE_STRINGS = 1;
```

This will make EmailPlus include all string fields discovered in records at any depth of recursion. However it is up to the administrator to ensure they only select fields that contain lists of valid ClearQuest Userids that are specified one per line or separated by commas. The results are otherwise undefined.

These constants should be defined in the PERL Global Script 'EMP_Customisable' with GLOBAL SCOPE i.e. not inside a function or subroutine. See the code extract below for an example of this.

```
# Start of Global Script EMP_Customisable
# The routines in the module are customisable.

$MAX_FIELDPATH_RECURSE_DEPTH = 2;
$FIELDPATH_INCLUDE_STRINGS = 1;

sub EMP_SubstituteCustomMetaTag {
    my $metaTagParts_ref = shift; # a reference to an array
```



```
my $entityDefObj    = shift;  
my $isHTMLContent  = shift;  
my $isTest         = shift;  
... .
```

If these constants are not defined then the package will adopt its default behaviour of:

- Only using User fields of type *Reference To Users* to determine email addresses, and
- Only searching to a depth of recursion equal to 2, i.e. child and grandchild records of the current record in order to determine the fields that can be used to determine email addresses.

Appendix E – Example EmailPlus Template Expressions

Included below are some examples of valid expressions that can be used in EmailPlusTemplate records to reformat information retrieved from ClearQuest or enhance the content of the email notifications:

- This expression will add HTML code to change the colour of the text depending on the value of the priority field:

```
#@EXPRESSION::if ( #?Priority?# eq "1 - High" ) { "<FONT COLOR=\"red\">"; } elsif ( #?Priority?# eq "2 - Medium" ) { "<FONT COLOR=\"orange\">"; } elsif ( #?Priority?# eq "3 - Low" ) { "<FONT COLOR=\"yellow\">"; } @#Priority: #?Priority?# </FONT>
```

For example, if the priority has a value of 1 then the text will be displayed in red, i.e. **Priority: 1-High**.

- This expression will print all of the record ids for the related records on the same line separated by spaces. If there are no related records associated with the record then nothing will be printed:

```
#@EXPRESSION::if ( scalar( @{#?RelatedRecords?#} ) ) { "Related Records: ".join( " ",@{#?RelatedRecords?#} ); } else { ""; }@#
```

For example, Related Records: mydb00001201 mydb00001543 mydb00002175

- This expression uses two Perl modules to manipulate the date string that is returned from ClearQuest to allow you to modify the output format of the date. It uses the time2str function in the Date::Format Perl Module and the str2time function in the Date::Parse Perl Module:

```
#@EXPRESSION::use Date::Format;use Date::Parse; my $template = "%d-%m-%Y %H:%M:%S";  
time2str( $template, str2time( #?SubmitDate?#, "UTC" ) );@#
```

For example, this expression would transform the date string value of the SubmitDate field “2009-10-20 16:22:40” to the UK formatted date string : “20-10-2009 16:22:40”

References : <http://search.cpan.org/~gbarr/TimeDate-1.19/lib/Date/Parse.pm>
<http://search.cpan.org/~gbarr/TimeDate-1.19/lib/Date/Format.pm>

- This expression will print the current value of the "State" field and if it has changed it will also print the old value of the "State" field:

```
State: #?State?# #@EXPRESSION::if ( #?State?# ne #?State?# ) { "(Old Value: ". #?State?#  
." )"; }@#
```

For example, State: Activated (Old Value: Submitted)

Appendix F – Example EmailPlusRule Advanced Rules

This appendix includes examples of some expressions that you can use in the EmailPlusRule Advanced Rules to define or enhance your rule conditions:

- This expression will return TRUE if the value of the SubmitDate field is within the last 24 hours.

```
StoDT( Gfv( "SubmitDate" ) ) > ( time() - ( 24*60*60 ) )
```

Function Gfv() retrieves the value of SubmitDate in UTC Date/Time format (for example 2009-08-16 09:17:54).

StoDT converts the UTC Date/Time into an elapsed time in seconds (i.e. 1250414274).

Time() returns the current elapsed time (for example 1250439474).

This elapsed time is reduced by 24 hours represented in seconds (24*60*60).

If the time now is 7 hours later than when the record was submitted:

StoDT("2009-08-16 09:17:54") > (1250439474 – 86400) evaluates to
(1250414274 > 1250353074) which evaluates to 1 (TRUE).

- This expression will return TRUE if the value of the Project.Name field contains the case insensitive word “test” :

```
( Gfv( "Project.Name" ) =~ /test/i )
```

For example, if the name of the project associated with the modified record is “Project Omega” then:

(“Project Omega” =~ /test/i) evaluates to 0 (FALSE).

For example, if the name of the project associated with the modified record is “My Test Project” then:

(“My Test Project” =~ /test/i) evaluates to 1 (TRUE).

Appendix G – Debugging EmailPlus

To debug EmailPlus operations

- Define a system environment variable called EMP_DEBUGOUT_LEVEL and set the value to one of the following integer values:
 - 1 - List all EmailPlus function calls and the parameters passed to the functions,
 - 2 - List all EmailPlus function calls, the parameters passed to the functions and the return values, or
 - 3 - Full Debugging – This can generate a lot of output.
- If you are debugging on Linux/UNIX you must also set the value of the EMP_DEBUGOUT_FILE system environment variable to the name of a file that ClearQuest can access and write to. You should not include any spaces in full path of the name of the debug file e.g. EMP_DEBUGOUT_FILE=/tmp/EmailPlusDebug.log.

To view the debug output on Windows run the Windows Debugger (dbwin32.exe). The debug output generated by EmailPlus will be prefixed by:

```
#EMP_DEBUGOUT# <DATE> : <FUNCTION NAME>: <DEBUG MESSAGE>
```

For example, when EMP_DEBUGOUT_LEVEL is set to 2 following output may be generated:

```
3712: #EMP_DEBUGOUT# 2009-10-23 11:45:21 :EMP_isEmailPlusActive: START
3712: #EMP_DEBUGOUT# 2009-10-23 11:45:21 :EMP_CacheConfig: START - Cache EmailPlusConfig
3712: #EMP_DEBUGOUT# 2009-10-23 11:45:21 :EMP_CacheConfig: END
3712: #EMP_DEBUGOUT# 2009-10-23 11:45:21 :EMP_isEmailPlusActive:
EMP::ConfigCache::isEmailPlusActive = <YES>
3712: #EMP_DEBUGOUT# 2009-10-23 11:45:21 :EMP_isEmailPlusActive: END - isEmailPlusActive=<1>
```

To view the debug output on Linux/UNIX view the contents of the file that you defined in the EMP_DEBUGOUT_FILE system environment variable.

Enabling debugging for EmailPlus will impact the performance of ClearQuest. It should only be enabled when you need to debug something and then disabled when complete. Also, you must define or modify the system environment variable before starting the ClearQuest client. Any changes made to the system environment variable while the ClearQuest client is running will not be recognised until the ClearQuest client is re-started.

©Copyright IBM Corporation 2007, 2009

IBM Corporation
Software Group
Route 100
Somers, NY 10589
USA

IBM, the IBM logo, Rational, the Rational logo, Telelogic, the Telelogic logo and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both.

Other company, product, or service names may be trademarks or service marks of others.

ALL RIGHTS RESERVED.

The Rational Software home page on the Internet can be found at
ibm.com/software/rational

The IBM home page on the Internet can be found at **ibm.com**