Email Functions

Email functions allow messages to be sent from scripts either as a predefined project or system email or by a complete specification of options.

Validating Email Addresses

To validate an email address, you can use a regular expression to test that an email address follows specific rules. Check if it matches (the `RegExMatch()` function returns greater than 0 matches) and if so, return "valid". The rules are that the address consist of:

- a name composed of specific characters (alphanumeric, period, underscore, percent, plus, hyphen), followed by
- an @ symbol, followed by
- a domain name composed of specific characters (alphanumeric, period, hyphen), followed by
- a single period, followed by
- a top-level domain name composed of alphabetical characters, from two to four characters in length.

The entire expression is enclosed in parentheses to create a marked sub-expression; if it matches, "1" will be returned.

Note that the backslashes in the regular expression string need to be escaped by being doubled.

```
If(RegExMatch(emailaddress, "([A-z0-9._%+-]+@[A-z0-9.-]+\\.[A-z]{2,4})") >
0,
    "valid",
    "invalid");
// An email address "first.last@example.com" will return "valid"
```

For additional information, see String Functions and the function `RegExMatch()`.

SendEmail

Declaration

```
string SendEmail(string from, string to, string subject, string message[,
string smtpServers, string account, string accountPassword, string cc,
string bcc, string replyTo, bool useSSL])
```

Syntax

```
SendEmail(<from>, <to>, <subject>, <message>[, <smtpServers>, <account>,
<accountPassword>, <cc>, <bcc>, <replyTo>, <useSSL>])
```

Required Parameters

- **`from`:** The email address to be used for the "from" field
- **`to`:** A comma-delimited list of email addresses to send the email message to
- **`subject`:** The subject of the email message
- **`message`:** The message of the email message

Optional Parameters

- **`smtpServers`:** A comma-delimited list of SMTP servers in the form of *server:port*, such as `smtp.example.com:465`. Port is optional, and standard SMTP ports will be tried if it is not included. The first successful server will be used.
- **`account`:** Account name to be used at the SMTP servers
- **`accountPassword`:** Password for the account at the SMTP servers
- **`cc`:** A comma-delimited list of email addresses for sending a copy of the email to
- **`bcc`:** A comma-delimited list of email addresses for blind "carbon-copying"
- **`replyTo`:** An email address for the reply-to field
- **`useSSL`:** Boolean flag indicating if SSL/TLS is to be used when authenticating with the SMTP servers.

Search in This Topic

On This Page

Related Topics

- Formula Builder
- Formula Builder Debugging
- Formula Builder Features
- Using the Formula Builder

Last updated: Apr 17, 2019

NOTE: Multi-byte characters can be used with any of these parameters as of Harmony Agent version 10.1.

Description

Sends an email using the supplied information. On success, an empty string is returned; otherwise, any error messages are returned.

Only the first four parameters are required; the rest are optional. If information is not specified, it will be read from the server configuration file. If it is not available in the configuration file, the email will not be sent.

Examples

Example 1

```
// Sending an email
// From address
// To address
// Subject
// Message
// SMTP servers
// Account
// Account password
// CC list
// BCC list
// Reply-to address
// Using SSL
SendEmail("john.doe@example.com",
    "jane.doe@example.com,firstname.lastname@example.com",
    "Weird data in PO input",
    "There was some weird data in this incoming PO, could you have a look
at it?",
    "http://mail.example.com",
    "john.doe@example.com",
    "mymailpwd",
    "jennifer@example.com",
    "svennis@example.com, urban@example.com",
    "jd@example.com",
    1);
```

Example 2

```
// Sending email, showing port configuration
SendEmail("myname@example.com",
    "othername@example.com",
    "Sendmail TEST",
    "A Sendmail TEST message.",
    "smtp.example.com:587",
    "myname@example.com",
    "mypassword", '', '', '', 1);
```

In the last example, an SMTP server listening on port is used. As there are no addresses being copied or blind copied on the email message, they are shown as empty strings. As there is not a "reply-to" email address, it is also an empty string.

[Back to Top]

SendEmailMessage

Declaration

```
string SendEmailMessage(string emailMessageId)
```

Syntax

```
SendEmailMessage(<emailMessageId>)
```

Required Parameters

- **`emailMessageId`:** An email message in the current project

Description

Sends an email using a predefined email message. On success, an empty string is returned; otherwise, any error messages are returned.

The email message used in this function call must be defined as an email message in the current project. See the instructions on inserting project items.

Examples

```
// Send a predefined email message
SendEmailMessage("<TAG>Email Messages/My Email Message<TAG>");
```

[Back to Top]

SendSystemEmail

Declaration

```
string SendSystemEmail(string to, string subject, string message)
```

Syntax

```
SendSystemEmail(<to>, <subject>, <message>)
```

Required Parameters

- **`to`:** A comma-delimited list of email addresses to send the email message to
- **`subject`:** The subject of the email message
- **`message`:** The message of the email message

Description

Sends an email using a pre-configured "From" address, SMTP servers, and account information. These are defined in the server configuration file. If these have not been defined, use one of the other email functions. On success, an empty string is returned; otherwise, any error messages are returned.

Examples

```
SendSystemEmail(Get("manager_email"),
    "FYI",
    "About to process a new purchase order with POID = " +
        Get("POID") +
        ". Will be available for review in a few minutes.");

// Examples entries in a jitterbit.conf file for sending system emails:
[SmtpClient]
DefaultSmtpServers=mail.example.com
DefaultFromEmail=auser@example.com
# If your SMTP server needs authentication the account and password can be
specified:
DefaultAccount=auser
DefaultAccountPassword=apassword
# If your SMTP server uses SSL/TLS, set this to true:
UseSSL=true
```

[Back to Top]