InAlias v 4.4
May 6, 2013

InPress
Systems

CONTENTS

1 WHAT IS INALIAS

InAlias is a software package for the Xinet WebNative Suite. It adds valuable functionality concerning User Registration, User Management and Login services.

InAlias works in both WebNative and WebNative Portal environments. Funcionality may differ slightly depending on the environment. For example, there is no login form for Portal since that is a function of Portal already.

1.1 User Registration

The main benefit of InAlias is to offload the server manager from the tedious task of creating user accounts. Users may register to the server themselves and the creation of new accounts can be done by InAlias automatically.

The account registration and creation process can be configured in different ways, from completely automatic to approval by site administrators using email notifications with links.
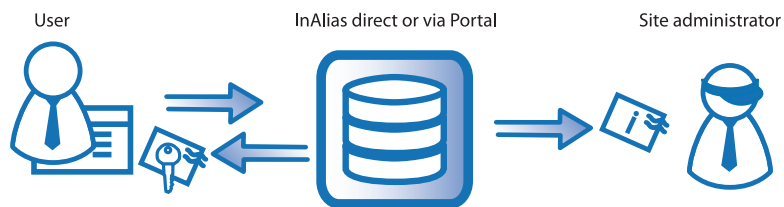
User registers and automatically gets an account. Siteadmin is notified
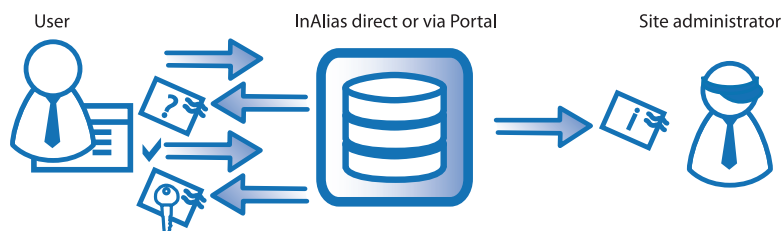
User registers and gets an email prompting to confirm the request. Account is created after confirmation. Siteadmin is notified
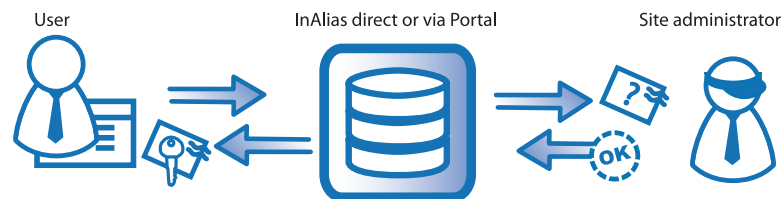
User registers and an email is sent to the siteadmin who can approve or reject. Account is created after approval.
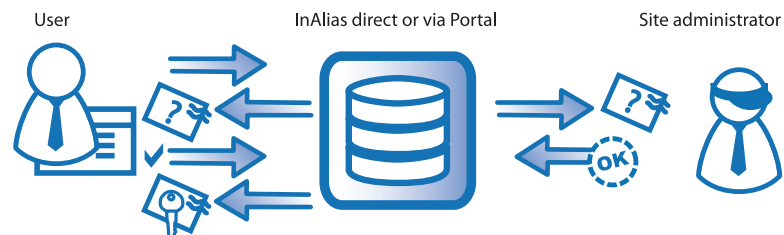
User registers and gets an email prompting to confirm the request. An email is sent to the siteadmin who can approve or reject. Account is created after approval

Accounts are created by InAlias using template setups created by the server administrator, thus keeping strict control on what the user access and permissions will be and at the same time keeping things simple to the user who registers.

The InAlias interface can be used directly by the registering user or as a simple subadministrations tool that can be handed over to external administrators that need no access to the main administration interface.

User selfregistering in an InAlias site from Portal

InAlias as a subadmin solution embedded inside an InPressive site

1.2   User Management

InAlias has the ability to work with expiration of accounts, keeping track of time since registration and notifying the user and/or administrator when an account is about to expire. An expired account is locked and access denied until the account is extended.
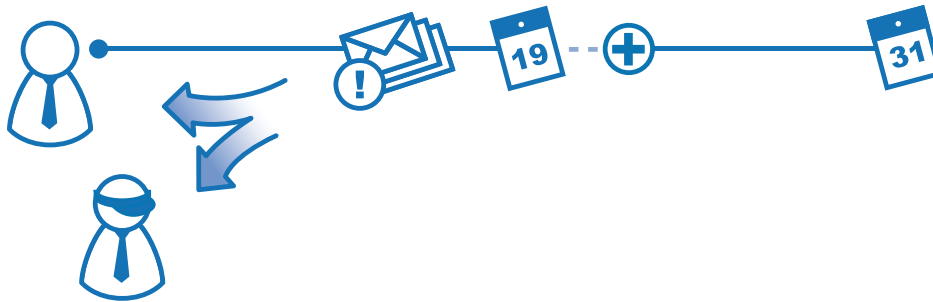
An account is created with an expiry date. InAlias sends notifications to user and admin when the account is about to expire, making it possible to extend the account.

When registering for an account, additional data can be collected to be used by administrators. The information, like phone numbers and addresses, is stored in the InAlias database and can be retrieved through InAlias or from other InPress Systems´ products.

Metadata can be collected while registering

1.3   Login services

InAlias can also act as a replacement for the http authentication popup window in WebNative. By making an InAlias site with integrated landing page, a nicer and more user friendly approach of username and login can be achieved.

Each InAlias site has its own look and feel and is accessed through its own url.

With InAlias it is also possible to create links that will autolog a user into WebNative. Just clicking the link will take care of the authentication and log a user into a specific account using a predefined username and password. With this technique it is possible to create "open" WebNative sites that need no authentication by the accessing user.

2 INSTALLING INALIAS

2.1 Platform requirements

InAlias is available for MacOSX Intel, RedHat Linux 5 (32-bit), RedHat Linux 6 (64-bit) and Solaris 10+.

2.2 Where do I get the software

To get the latest InAlias software, go to ftp.inpress.se, login using "anonymous" and provide your e-mail address as a password. As an alternative, the installation package can also be provided to you by an InPress Systems reseller or from InPress Systems support.
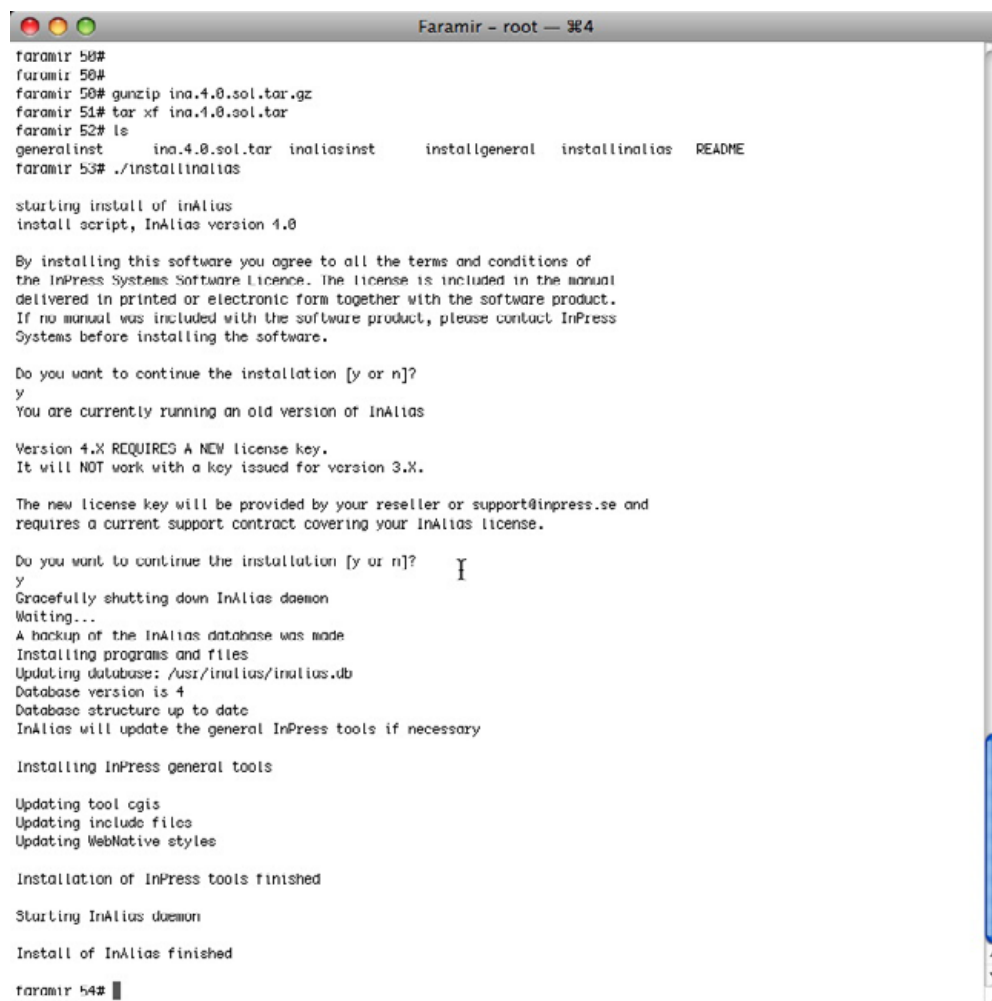
The installation package is a gzipped unix tar file named "ina" followed by the version and platform and ending in "tar.gz". For example ina.4.4.intosx.tar.gz refers to InAlias version 4.4 for the Mac OSX operating system on Intel processors.

2.3 Installing the software

Login to the server as root and place the software in an appropriate installation directory. Uncompress and untar the file to display the **inaliasinst** folder, a **README** file and an installation script: **installinalias**.

Run the script by typing *./installinalias* and press return.

Any current installation will be upgraded to the version you are installing.

Running the installation script on a system where InAlias is already installed

You need to be logged in as root when installing the software

2.4 Updating the Apache configuration

The InAlias software is mainly installed in /usr/inalias. The installer attemps to update the apache http.conf file to add aliases and directories to access the inalias cgi which needs to be accessed without authentication.

The addition of the configuration in the apache config can be made manually in case the automatic update does not work. The following section is inserted in the conf file by the installation (it can be found within the /usr/inalias/setup folder as well):

```
# InPress InAlias config - DO NOT REMOVE

ScriptAlias /inalias /usr/inalias/cgi
Alias /inaliasdoc /usr/inalias/web

# InAlias directory
<Directory /usr/inalias/cgi>
AllowOverride None
AuthType Basic
AuthName inAlias
Options FollowSymLinks ExecCGI
Order deny,allow
</Directory>

<Directory /usr/inalias/web>
AllowOverride None
Options FollowSymLinks
Order deny,allow
</Directory>

# END InPress InAlias config - DO NOT REMOVE
```

2.5 Starting up

The installation script installs the proper startup scripts for all platforms and then starts the daemon. The daemon should start and stop using the normal startup procedures.

In case you need to stop the daemon manually, run the following from the command line

```
# /usr/inalias/bin/inalias_ctrl stop
```

In case you need to start the daemon manually, run the following from the command line

```
# /usr/inalias/bin/inalias_ctrl start
```

3 CONFIGURING INALIAS

The web-based InAlias admin utility can be found on the plugins tab in the Xinet nativeadmin tool:

It can also be accessed from the web address

```
http://<yourserver>/webnative/plugins/inaliasadmin
```

You need to be logged in as "nativeadmin" to access the admin utility. It is organized the same way as the other admin utilities for InPress Systems, with tabs for different functions and links for licensing and base configurations.

Main InAlias admin screen

3.1 Licensing

InAlias needs a 16 digit license string provided by your reseller. The string is entered in the area for licensing and after clicking the button **Update** you will see what kind of license you have purchased.

To provide the license, your reseller needs the Hardware ID displayed above the entries for serial number and license string. The serial number is also provided by the reseller.

Licensing of InAlias

3.2 Working with sites

All InAlias configuration is organized into *Sites*. The site can be a full blown registration solution or a simple autologin link. Each site is associated to an *Access Code*. The access code is used to access the specific landing page in the WebNative environment or to connect a Xinet Portal site to a certain InAlias configuration when used via Portal.

All users that register through InAlias will end up in a specific site and can be displayed in the site´s userlist.

Each site also has a name. The name is used internally by InAlias while the accesscode is used to access the site from the outside. The name of a site cannot be changed after it has been created.

To create a new site or to find a site to work on use the SITES tab. The page lists all the sites that are availalable on the server. The list is the starting point to get to the administration of a site.

If the list is long, it is possible to type into to the input field above the list to filter the list to sites that include the string that has been typed. The typed string will search both in the sitename and the access code shown within paranthesis.

When a site has been selected (by clicking or filtering), some base information will be shown about the site to the right side of the list.

Site listing

Type in a string that will be used to narrow down the list

If you know a certain username, it is possible to lookup the site where that user has been registered. Type the name into the input below the list and use the Lookup user button to find the site. If the site is found it will be automatically selected. If the user is not found in InAlias, an error will be displayed.

Use a username to find the site

3.3   Creating and editing a site

All InAlias configuration for a particular use - no matter if it's a full blown registration solution or a simple autologin link - is associated to a site. Each site has a name and an accesscode. The name is used internally by InAlias and the accesscode is used to access the site from the outside. The name of a site cannot be changed after it has been created.

A sitename cannot be changed after it has been created

So, if you for example need to create a registration workflow for one client, a different registration workflow for another client and an autologin setup for accessing a Portal site with "one click" you would need three different sites.

To access a site when using WebNative, the accesscode is added to the inalias url:

        http://SERVERNAME/inalias/inalias?ACCESSCODE.

Depending on the site configuration, this url will either access a startscreen or initiate the autologin feature.

Example: If your webnative server has the address
        http://webnative.inpress.se
and you created a site with the accesscode abc123, the address to the site would be
        http://webnative.inpress.se/inalias/inalias?abc123

The sitecode used

The URL to get to the site
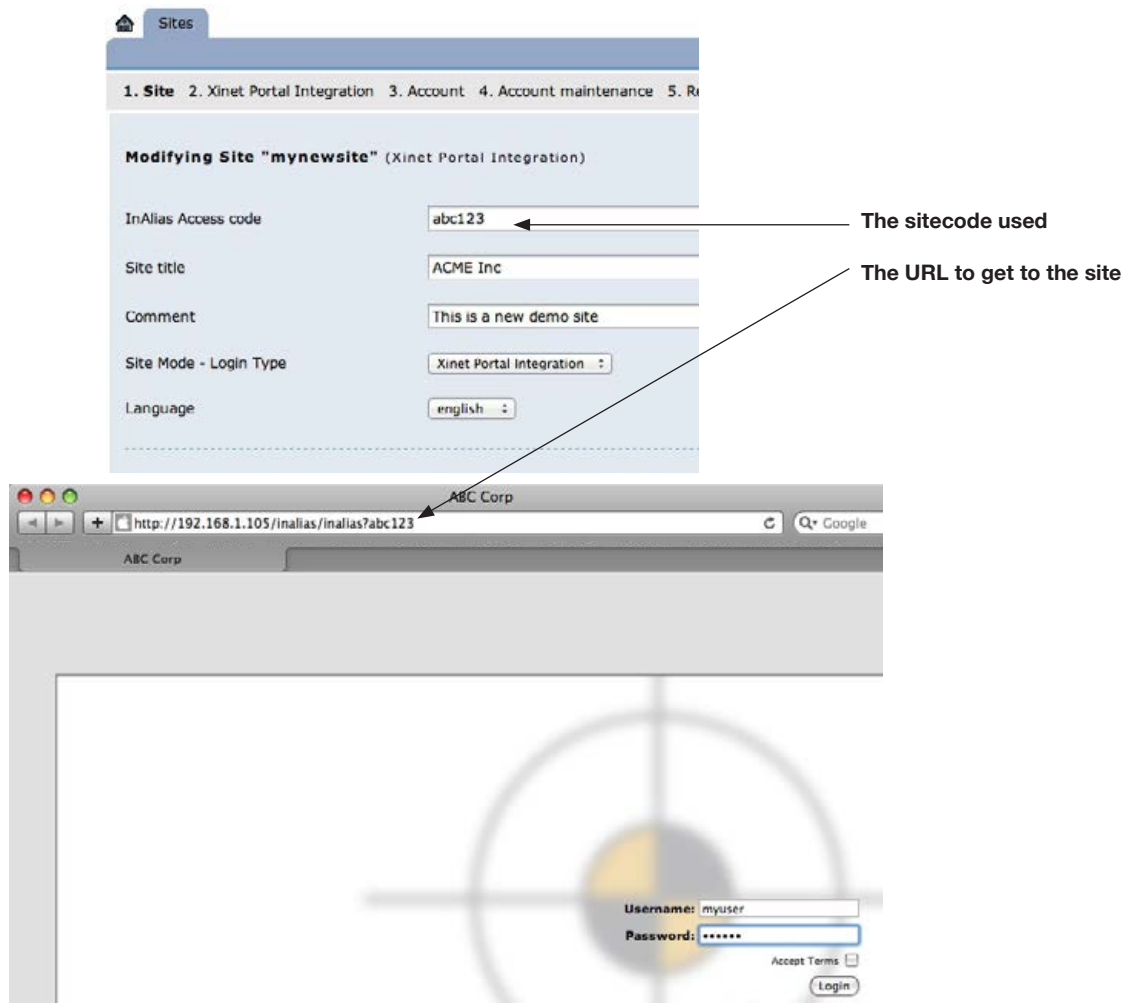
The relation between the sitecode and the URL to access the site

When InAlias is used with WebNative Portal, the configuration is similar but access to InAlias registration features is done through the normal Portal interface. Each Portal site needs to be configured to match a site in InAlias. The match is done using InPress Portal Integration Kit's configuration script. When the script is run, it will ask for the site access code in order to make the match.

InAlias access code is matched to the Portal site when running the configuration script

Note that this behaviour has changed with InPress Portal Integration Kit v 4. For more information about integrating InAlias with Portal, see *6 Using InAlias with Xinet Portal.*

To add a new site, go to the SITES tab and click **Add site**. Give the site a name, a Site access code that will be used on the URL to access the site (see above), a Title that will be shown in the registration forms, a comment for the admin and select the Site mode.

The name of the site is limited to using the characters a-z, A-Z as well as numbers, undescore (_) and dash (-).

When adding a new site, it can be based on an existing site. All files added to the site (custom pages and mail, and other setup files) will be copied when making a new site.

Adding new site to InAlias. The sitename cannot be changed after site has been created.

When a site has been added, you will automatically be redirected to the site´s full configuration screen of a site. The configuration of a site consists of several tabs, controlling different aspects of a site. Depending on the purpose and required operation of the site, different tabs will be used. Each tab is explained in detail in the following subchapters.

Note that some settings are not available for all Site modes.

The tabs may look different depending on the selected Site mode

3.3.1 Editing site, Tab 1 - Site

Tab 1 contains the base settings for the site including the information given when creating the site. In addition, here is where language is selected, as well as base email addresses that are needed and the access urls that are used in all email correspondense (links to confirm, approve and so on).

From version 4.4 it is also possible to temporarily close the site. This is useful when doing maintenance, etc. Closing the site will remove login for WebNative and prevent login when using Portal. No registration can be made while closed.

SETTING	DESCRIPTION
InAlias Access code	Code used to access site. Used in link to site: http://server/inalias/inalias?site_code When using InAlias with Xinet WebNative Portal, the access code is used to connect the InAlias site to a Portal site. This is done by running the configuration script (or editing the file inalias_conf.php in the Portal site)
Site title	Title of site. Available in the form pages and email templates as tag: #user.company#
Comment	A comment for the site. This information is for administration purposes and is not shown anywhere in the end user interface.
Site Mode - Login Type	Type of site/login. **Auto login** - use to create Automatic login via link **Form login** - use for creating a Login form **Version 2 login** - old type of login for InAlias version 2. Here for backwards compatibility **Xinet Portal Integration** - using InAlias from a Xinet Portal server **Xinet Portal Autologin** - automatic login in a Portal site
Language	Select the language for the site. Custom languages can be created by copying an existing language file inside /usr/inalias/setup/lang to a new name. A new file will automatically show up in this pulldown menu and be selectable.

Administrator E-mail	E-Mail address to the site administrator This E-mail address is used for notifications when an account is auto-created or requested (using approval) and for alerts on the Expiry functions. Multiple comma-separated addresses can be used.
Reply-to E-mail	Return E-Mail address in mails sent by InAlias using this site. This is a required entry if using emails.
External WebNative server URL *(not for "auto" types)*	This URL will be used to create the links for login, confirmation and approvals that are passed to the user or siteadmin in a email. When accessing InAlias from a WebNative server (site mode = Form login of Version 2 login) insert the URL to the WebNative host, for example: http://192.168.1.100 or http://mywebnative.company.com When accessing InAlias from a WebNative Portal server (site mode = Portal Integration) insert the URL to the Portal site where the InAlias config is being used. Use notation with /Site or a virtual host name if appropriate, for example: http://myportalserver.abc.com/Site or http://myportalsite.abc.com See more about integration with WebNative Portal in *6 Using InAlias with Xinet Portal*
Internal WebNative server URL *(not for "auto" types)*	This URL will be used to create a second set of links for login, confirm and approvals that are passed to the user or siteadmin in a email (not to the user making the request). This set is referred to as internal and is intended to be used for addresses inside a firewall when needed. When accessing InAlias from a WebNative server (site mode = Form login of Version 2 login) insert the URL to the WebNative host, for example: http://192.168.1.100 or http://mywebnative.company.com When accessing InAlias from a WebNative Portal server (site mode = Portal Integration) insert the URL to the Portal site where the InAlias config is being used. Use notation with /Site or a virtual host name if appropriate, for example: http://myportalserver.abc.com/Site or http://myportalsite.abc.com To override the logintype setting, add *;portal* or *;webnative* after the url. Example: 192.168.1.100;webnative. See more about integration with WebNative Portal in *6 Using InAlias with Xinet Portal*
Close site *(not for "auto" types)*	Close a site temporarilly.
Closed site message *(not for "auto" types)*	Message to user trying to access site.

1. **Site** 2. Xinet Portal Integration 3. Account 4. Account maintenance 5. Registration control 6. Reports & Email 7. Administration 8. Advanced

Modifying Site "mynewsite" (Xinet Portal Integration)

InAlias Access code	abc123
Site title	ACME Inc
Comment	This is a new demo site
Site Mode - Login Type	Xinet Portal Integration
Language	english
Administrator E-mail	admin@acme.net
Reply-to E-mail	noreply@acme.net
External Portal server URL	portal.acme.net
Internal Portal server URL	192.168.1.111;webnative
Close site	☐
Closed site message	

[Dismiss] [> Site/User Info] [> Manage Templates] [Save] [Save and Proceed >>]

Tab 1 contains basic setup items for the site

3.3.2 Editing site, Tab 2 - Site mode

Tab 2 contain setups very specific to the site Mode that has been selected. Depending on the selection of type, different fields will appear. Some fields only show for some types. The title of the tab will reflect what Site mode that has been selected.

The Site Mode setting is important to determine how InAlias responds to access and also for the admin utility to show the correct settings. Tab 2 will be completely different depending on the selection and some of the following tabs as well. Most settings do not make sense when the Autologin mode is used and for that reason, are not shown. The other three modes share most of the options on tabs 3-8. The rest of this subchapter shows tab 2 after making the selection of the mode.

When using **Auto Login** The site will login to WebNative without prompting for a username and password. Username and Password from the site config will be used. **Xinet Portal Autologin** will autologin from Xinet Portal.

It is also possible to configure InAlias to create users automatically if needed. See *7.2 Working with Autologin through links* for more information. This function does not work when using Portal.

SETTING (AUTOLOGIN)	DESCRIPTION
User ID	The User ID that will be used when logging into WebNative. The userID can also be passed on the url if dynamic users are needed (see Make autouser below). In that case, the passwords to be used should be kept in the InAlias database or in a lookup file inside the site folder. See *7.2 Working with Autologin through links* for more information.
Password	The Password that will be used when logging into WebNative. When using dynamic user ID´s, the passwords to be used should be kept in the database or in a lookup file inside the site folder. See *7.2 Working with Autologin through links* for more information.

Match Referrer	Only login when Referrer contains this string. The referrer is the location the user is coming from when accessing the link. Example: *inpress.se* will match any link accessed from an url *containing* inpress.se (for example www.inpress.se), and allow the login to happen.
Make autouser if needed	Dynamic users can be used with the link: http://server/inalias/ inalias?SITECODE+-u+USERNAME When using this option, InAlias can create the user if it does not already exist. Note that password and other data must be defined in a site autouser file. See *7.2 Working with Autologin through links* for more information. Other options for creating the user should be set in the User creation section below.
Add prefix to dynamic username	Add this Prefix to the username as passed on url when using a dynamic login username. Note that the full name including any prefix and suffix needs to be in the site autouser file.
Add suffix to dynamic username	Add this Suffix to the username as passed on url when using a dynamic login username. Note that the full name including any prefix and suffix needs to be in the site autouser file.
Login redirect path	Navigate to this path after login. Specify a full path to a folder where to go after login.

Tab 2 when selecting Sitemode Auto Login

Autologin link from InPress Systems website. Clicking link opens a window where InAlias logs the user in automatically

Autologin link using InAlias

When **Form Login** is the selected mode, the site will display as a login page with two form fields for supplying username and password.

The basic look and feel can be changed using the setups that follow

SETTING (FORM LOGIN)	DESCRIPTION
Background Image	Type in the urlpath to an image to use as background to the form. The image will be centered on the page. The url path */inaliasdoc* points to the folder */usr/inalias/web*. InAlias ships with the sample images *bgscreen.jpg* and *defaultbg.jpg* The default is: */inaliasdoc/defaultbg.jpg*
Image Width	The width of the selected background image. *Default 800*
Image Height	The height of the selected background image. *Default 450*
Background Color	Background color on the login page and on all the registration and other InAlias pages. Note that using a dark background color may make the labels in the forms unreadable. The color can be specified using hex colors like *#E3E9F0* or named colors like *silver*.
Box Color	When specified, a frame around the background image will be printed using this color. The color can be specified using hex colors like *#E3E9F0* or named colors like *silver*.
Login position	The login form and links to registration and such are placed on top of the background image. Using this control it is possible to select where to place form fields and links.
Only Login	Only use the site as a form login. Do not show any links to registration or other services
Open New Window	Open WebNative in new window when you login from InAlias.
Legal Text Label	Legal text label to show at login together with a checkbox that needs to be checked before logging in. This text will be a link to legal text if such a text is supplied. Leave this field blank if no legal or legal text is needed.
Legal Text	Legal text to show if the user clicks on the legal text label.
Login redirect path	Navigate to this path after login. Specify a full path to a folder where to go after login.

Tab 2 when selecting Site Mode Form Login

Form login using a legal text label and a legal text.

The **Version 2 Login** type is kept for backwards compatibility. The site will show login and registration links as images that can be specified below. Login is a normal link that will open the standard authentication dialogue for http authentication.

SETTING (V2 LOGIN)	DESCRIPTION
Login Image	Web-path to an image to use as the Login button. The url path */inaliasdoc* point to the folder */usr/inalias/web*. InAlias ships with the sample images login.gif and register.gif. The sample image *login.gif* will be used if no image is specified.
Register Image	Web-path to an image to use as the Registration button. The url path */inaliasdoc* point to the folder */usr/inalias/web*. InAlias ships with the sample images login.gif and register.gif. The sample image *register.gif* will be used if no image is specified.

Tab 2 when selecting Site Mode Version 2 Login

The **Xinet Portal Integration** mode is to be used when Xinet Portal is integrated with InAlias.

SETTING (PORTAL INTEG)	DESCRIPTION
Background Color	Background color on the registration and other InAlias pages. When intergrated with Xinet Portal, this is the color on the overlaying registration page and related pages. The basic site in Portal is not affected. Note that using a dark background color may make the labels in the forms unreadable. The color can be specified using hex colors like *#E3E9F0* or named colors like *silver*.
Panel Width	Input the panel width to use for the InAlias overlay using InAlias on the Portal Server. Default: 450
Panel Height	Input the panel height to use for the InAlias overlay using InAlias on the Portal Server. Default: 450
Restrict logins to active users	If this checkbox is checked, any user trying to login from a Portal site that has been configured to connect to this InAlias site will be stopped from login in. Only active users in the site (and users specified below) will be accepted.
Error message	Error message to display if login is refused using the Restrict logins to active users function.
Additional users when restricting login	The usernames listed will be treated as active when deciding wether a user has access to the site or not. Note that the users listed not need to be assigned to the InAlias site. List multiples commaseparated but without any extra space.
Additional groups when restricting login	The groupnames listed will be treated as active when deciding wether a user has access to the site or not. Each user attempting to login will be checked and if it is in one of the listed groups, it will be accepted. List multiples commaseparated but without any extra space.

Tab 2 when selecting Site Mode Xinet Portal integration

The **Xinet Portal Autologin** mode is to be used when Xinet Portal is integrated with InAlias to take advantage of the autologin capabilites. When used this way, the Portal site will automatically login using the user credentials given in the setup.

SETTING (PORTAL AUTOLOGIN)	DESCRIPTION
User ID	The User ID that will be used when logging into WebNative. It is not possible to pass the username on the URL like when using Autologin for WebNative only.
Password	The Password that will be used when logging into WebNative.

Tab 2 when selecting Site Mode Xinet Portal integration

3.3.3 Editing site, Tab 3 - Account

Tab 3 is used to decide how to handle registration, user creation and collecting additional meta-data fields.

InAlias has the ability to collect and restore additional data associated with a user. This data can be used from other InPress Systems products as well as searched and displayed in InAlias. The form for entering the fields shows one empty metadata field when starting and one empty field as soon as one has been added.

- Add new fields by typing into the empty form and clicking the + button.
- To edit an existing field, change and save the page.
- To delete a field, click the - button

Tab 3, setting up how to create users and what additional data to request.

SETTING	DESCRIPTION
Disable user registration	Disables the link for user registration Registration can still be made from the site/user info page
User creation steps	The steps needed to create a user can be selected from: **Automatically create user** - The user account is created automatically after registration **Confirm user by email** - A notification is sent to user who is requested to confirm the registration. The user account is created when the registration is confirmed **Request administrator approval through email** - A notification is sent to the site administrator who can approve or deny the request. The user account is created when the registration is approved **Confirm user by email and via administrator approval** - A notification is sent to the user to confirm, followed by a notification to the site administrator who can approve or deny the request. The user account is created when the registration is approved
User account creation method	The user account may be created using one of following methods: **Single user template** - The new user account is created by cloning the user specified in the User Template account name **Multiple user templates** - The new user account is created by cloning the user selected from the User/Group selection list. **Single user template with multiple primary groups** - The new user account is created by cloning the user specified in the User Template account name with a primary group selected from the User/Group selection list **Single user template + group input** - The new user account is created by cloning the user specified in the User Template account name with a primary group entered in input field. Group is created if needed.
User Template account name	Existing WebNative user account to use as the Template account that is cloned when creating a new user. *This setting will be ignored if the selected User account creation method is: Multiple user templates*
User/Group selection list	Selection list that is used to select a user template account or a primary group depending on what User account creation method is being used. Create the selection list by typing in titles and user/groups and clicking the + button.
Title to display	The title that should appear on the pulldown menu item.
Selected User or Group	User or group attached to the menu item. Note that the user account or group has to exist in WebNative.
Error Message for User/Group Selection	When using a Selection list to select user or primary group it is possible to have an empty value in the "Selected User or Group" selection in the list, for example to put in an instruction like: "Please select ...". By adding an error message, the selection becomes required and the error message entered is displayed if the user tries to submit without making a selection. Note that when using selection list to select Template User, the selection will always be required and a default error message displayed if no specific message is supplied.
Metadata Fields	Define fields for additional data to be collected when a user registers
Req (*)	Check to make the metadata field a required entry
Title	Title on field displayed in form
Name	Name of field in the form. This name is a regular form field name and cannot include high ascii or other special characters. The fieldname can be used to access the data from other applications

SETTING	DESCRIPTION
Type	Type of field. Currently the allowed types are: **Textfield** - A regular input field **Popup** - A popup menu built from typed in selections or a file **Textarea** - A multiline input field **Checkbox** - A checkbox field
Value(s)	Additional setting for the field. Used differently for different types: **Textfield** - Default value when registering **Popup** - popup values to display: *val|val|val* or */NAME* that point to a popup file from */inalias/site/SITENAME/popups/NAME* **Textarea** - Set size of field: rows="A" cols="B". Example: rows="8" cols="40" for a field 8 lines high and 40 characters wide. Default size is 5 rows and 35 columns. **Checkbox** - A value to set when the checkbox is checked. Default is "1"
Use for upload email	Select wether to populate the Upload email setting from the user registration
Use for User email	Select wether to populate the User email setting from the user registration

3.3.4 Editing site, Tab 4 - Account Maintenance

Tab 4 is used to setup the user maintenance options like Expiry of users and Password requests. It also holds the options for expiring accounts.

Tab 4 controls the way users are expired and how Password requests are handled.

SETTING	DESCRIPTION
Allow request for lost password	Select method how to handle password requests: - No password request - Send password in email - Reset password and send link for updating password

SETTING	DESCRIPTION
Notify admin on password request	Send an email notification to site admin when a user request a password.
No password question required	When checked, password can be retrieved without a password security question.
Password security questions	Enter several different questions that the user can select from when requesting an account. The selected question is saved along with the answer provided by the user when the request is saved. When password retrieval is used, the question is displayed and the user is prompted for the answer. The answer has to be correct in order for the password to be sent out.
Number of days before Expiration	Enter the number of days an account will be active before automatically expiring and be locked.
Allow user to set expire time	The registration form will include a field to set a dynamic expiration time when this option is checked. This option should only be used when InAlias is used by administrators as opposed to an open site for registration.
User notification and extend method	The user email alert type and method of extending account: **No alert email** - The user will get no E-mail alert and have no way to extend the expired account **User e-mail link** - The user will get one or more E-mail alerts with a link for extending the account. When clicking the link, the account will be extended. **User e-mail link & change password** - The user will get one or more E-mail alerts with a link for extending the account. When clicking the link, the account will be extended after the user has supplied a new and different password.
Send alert to administrator	Checking this option will tell InAlias to send alert E-mails for expiring accounts to the administrator of the site. The administrator E-mail will contain a link for extending the user account.
Alert Frequency	Configuration for the number of alerts to send and how long prior to expiration the alerts will be sent out. Enter the number of days before expiry that the alerts should be sent out. For multiple alerts, enter the different lengths separated by ; (semi-colon) Example: 5;3;1 - Alerts will be sent out 5, 3 and 1 day(s) before the expiry as well as right after the expiration takes place.

3.3.5 Editing site, Tab 5 - Registration control

Tab 5 is used to setup controls and custom options on how to handle username, password and email entry.

Tab 5 controls usernames, passwords and email entry and creation.

SETTING	DESCRIPTION
User ID input control	Select a filter to apply to input of user name. InAlias ships with the filters *default* and *email*. **default** - removes unwanted characters from the input **email** - requires a username formatted as an email address It is possible to add new filters by creating files that are placed inside */usr/inalias/setup/java*
Verify user ID against a file	Require the username to be in a list supplied as a textfile in the site. Enter the name of the verification file here. The file needs to be saved to the corresponding site folder - */usr/inalias/site/SITENAME* - as a tabseparated iso latin file with UNIX linefeeds. If the name of the file starts with / it will be interpreted as a full path and can be placed anywhere on the server. InAlias will search the first column in the file and allow only User Id´s that exist in the file.
Verify user ID with regexp	Entered user ID will be required to match a regular expression. Example: *.*@somedomain.com* will require username to end with @*somedomain.com*

SETTING	DESCRIPTION
User ID blacklist	The user ID will be tested against a list of forbidden words or phrases. The list can include wildcards like *root* or *abc*. Type in the name of the blacklist file to use. The file needs to be saved to the corresponding site folder - */usr/inalias/site/SITENAME* - as an iso latin file with UNIX linefeeds. To control the case sensitivity of the test, a second argument can be passed after the name separated by a ; (semicolon). The argument is either l (*lowercase*), u (*uppercase*) or c (*case sensitive*) and describes how the list should be interpreted. Lowercase is the default. Example: myBlacklistfile;l specifies to use */usr/inalias/site/SITENAME/myBlacklistfile* and to read the blacklist words as lowercase. The typed in user ID will be converted to lowercase before being compared to the list
Make Full Name entry required	Make the field for Full Name required for registration.
E-mail entry options	E-mail entry can happen in three ways: **No selection** - different entries will be used for user ID and email **Use User ID as email** - only one entry field will be used for both user ID and email **Make Email entry required** - different entries will be used for user ID and email. Email address is a required field
Verify E-mail with regexp	Entered E-mail will be required to match a regular expression. Example: *.*@somedomain.com* will require e-mail to end with *@somedomain.com*
E-mail blacklist	The Email will be tested against a list of forbidden words or phrases. The list can include wildcards like **@hotmail.com* or *info@**. Type in the name of the blacklist file to use. The file needs to be saved to the corresponding site folder - */usr/inalias/site/SITENAME* - as an iso latin file with UNIX linefeeds. To control the case sensitivity of the test, a second argument can be passed after the name separated by a ; (semicolon). The argument is either l (*lowercase*), u (*uppercase*) or c (*case sensitive*) and describes how the list should be interpreted. Lowercase is the default. Example: myBlacklistfile;u specifies to use */usr/inalias/site/SITENAME/myBlacklistfile* and to read the blacklist words as uppercase. The typed in user ID will be converted to uppercase before being compared to the list
Read E-mail from file	Read E-mail address from a textfile. InAlias will search the first column of the specified file to find the User ID. The E-mail will be picked up from the column specified below. The second column will be used if no column is specified. Enter the name of the lookup file here. The file needs to be saved to the corresponding site folder - */usr/inalias/site/SITENAME* - as a tabseparated iso latin file with UNIX linefeeds. If the name of the file starts with / it will be interpreted as a full path and can be placed anywhere on the server
Column to read in Email list file	Number of column (tab-separated) where E-Mail address is to be read (when user ID is matched). Default is 2
Password entry	Password entry can be handled in several ways: **Password not required** - input field for password is displayed, possible to register without a password **Autocreate password** - no password input, password will be created automatically **Autocreate password if left blank** - password will be created automatically if user leaves field blank **Password entry is required** - input field for password is displayed, user is required to enter password

SETTING	DESCRIPTION
Password blacklist	The Password will be tested against a list of forbidden words or phrases. The list can include wildcards in the words like *abc* or *qwert**.

Type in the name of the blacklist file to use. The file needs to be saved to the corresponding site folder - */usr/inalias/site/SITENAME* - as an iso latin file with UNIX linefeeds.

To control the case sensitivity of the test, a second argument can be passed after the name separated by a ; (semicolon). The argument is either l (lowercase), u (uppercase) or c (case sensitive) and describes how the list should be interpreted. Lowercase is the default.

Example: myBlacklistfile;c
specifies to use */usr/inalias/site/SITENAME/myBlacklistfile* and to read the blacklist words as case sensitive. The typed in user ID will be compared to the list as it is. |
| **Password input control** | Select a filter to apply to the user input of password.

InAlias ships with a default filter that checks the length of the password and use of special characters (not A-Z or a-z). The minimum number of characters in the password and any minimum number of special characters required should be specified in the Password filter parameter entry below.

It is possible to add new filters by creating files that are placed inside */usr/inalias/setup/pwfilters*. When adding custom filters, the parameters below can be used as needed. |
| **Password filter Parameter** | Parameters that are passed to the filter selected above.

The default password control filter shipped by InAlias expects two values separated by a semicolon. The first value is the minimum length of the password and the second (optional) value is the minimum number of special characters (ie characters that are not a-z or A-Z).

Example: 6;2 - The password must be at least six characters long and include at least two special characters
Example: 8 - The password must be at least eight characters long |
| **Password filter error message** | Error message that will be passed to the filter.

Using the default filter, the message will be shown in an alert if the password does not match the requirements.

Using custom password filters, the javascript in the filter needs to add support to this feature in order to be used. |

Blacklist files can be used to prohibit the use of certain words and character combinations in user ID´s, emails and passwords. To use blacklist for user ID, email or password, a blacklist file needs to be created for each one. The file should be saved to the site folder: `/usr/inalias/site/SITE-NAME`. The name of the file is entered in the corresponding blacklist entry.

A blacklist is a list of entries that can include wildcards. The entries should be listed one per line and the file should be saved as an iso latin file using UNIX line endings.

Example of entries in a blacklist file for emails:

```
*hotmail.com

*yahoo.com

info@*

sales@*
```

User IDs can also be verified using a lookup file. The full user IDs need to be listed in the first column of the file using tab as a column delimiter. It is also possible to use a file with only one position per line. The file needs to be an iso latin file using UNIX line endings. The file can be in the site folder or anywhere on the filesystem. Using a filename only will be interpreted as inside the sitefolder.

Email can be picked up from a file using the lookup with the user ID as the first column. It is possible to use the same file for Email pickup as for user ID verification.

Example of a listing of allowed User IDs and email addresses:

```
R12345          someone@somewhere.net

R23456          someone.else@somewhere.net

R34567          another@elsewhere.net
```

In this case, users would be verified to the file and emails picked up using the second column automatically.

3.3.6 Editing site, Tab 6 - Reports & Email

Tab 6 is used to setup time and date formats to be used in emails and reportings. It is also where to setup the use of running reports.

Tab 6 is also used to enable and control email broadcast settings for the site. Editing and using the Broadcast feature is explained in detail in *5 Broadcast*

The Broadcast feature is a way to batch send emails to all users registered to the site in question. It is based on the use of email templates. The feature is enabled by entering a *Broadcast Access Code*. This code is needed when a user accesses the feature from the site starting page or from Portal (using the InAlias integration). It is also possible to create a link for broadcast in InPressive styles.

The Broadcast feature uses E-mail templates that define the look and feel of the message as well as what information the user is required to input in order to generate the message. Making E-mail templates as well as other topics regarding the Broadcast feature are covered in *5 Broadcast*.

Tab 6 contains all the options concerning reports as well as Broadcasting Emails to site users

SETTING	DESCRIPTION
Report path	Specify the path where to save report The report path can be dynamic and contain the same variables as the report itself. It is also possible to include a date() command. Using the dynamic path, it is possible to create for example daily or weekly reports. Example: /raid/InAlias Reports/&user.siteName&/&ina-statgroup&.&date(%Y-%m-%d)&.xml can translate into something like: /raid/InAlias Reports/mySite/adduser.2012-02-28.xml For more information about reports, see "4 Customizing Pages, Emails AND REPORTS"
Report template	Layout of the report written to the report path. If there is a file already present, the report will be appended, otherwise it will be created. The name on the pulldown corresponds to a file inside the report base-folder */usr/inalias/setup/reports*. To create a new receipt/report template setup, copy any of the sample templates in the basefolder into a new name and edit. Select the corresponding name to use the template inside. For more information about reports, see "4 Customizing Pages, Emails AND REPORTS"
Limit reporting to	Check one or more event groups to limit reports to. If no report group is selected all events will be logged (if a template has been selected and a path specified) Requesting accounts : requests, confirms, cancelations, etc Adding accounts : creating, autoadding and approving accounts etc Account maintenance : extending, requesting passwords etc For more information about reports, see "4 Customizing Pages, Emails AND REPORTS"
Date format in emails	Select the date format to use in emails.
Time format in emails	Select the time format to use in emails.
Date & time format in emails	Select the date & time format to use in emails.
Broadcast Access Code	Access code for entering the Broadcast Email function. Typing in a password here enables the access to Broadcast. A user that wants to access this feature needs to supply this password. Broadcast is used to send out email batches to all users in a specific site. The E-mail content is decided by customizable templates with placeholders for unique data. For more information about creating templates see *5 Broadcast*
Send using BCC	see *5 Broadcast*
Maximum addresses per email	see *5 Broadcast*
Archive mail path	see *5 Broadcast*

3.3.7 Editing site, Tab 7 - Administration

Tab 7 is used to configure Administration options like editing and seraching for user data as well as allowing for deleting users.

Tab 7 - Controlling admin functions.

SETTING	DESCRIPTION
Metadata edit Access Code	Access Code for site administrator to edit metadata. This Access Code needs to be entered along with an existing username in order to edit data for a user.
User allowed to edit	Enable the ability for a user to edit metadata by giving his/her password.
Edit base data	Editing will also be possible for base data for a user: Full name, password, email and password question.
Search Access Code	This code will grant access to search for users and display metadata.
Erase account Access code	Authorization code needed in order to delete an account. Supplying this code along with an existing user ID that is registered to this site, will delete that user.
Select user ID	When checked, the site administrator using the delete function can select a user from the site instead of typing in the user ID. Note that when enabling this function it is possible to see the user names of the site. Use only in controlled environments.

3.3.8 Editing site, Tab 8 - Advanced

Tab 8 is used to configure some miscellaneous setups considered to be "advanced". Advanced options includes creating volumes when creating users and running external scripts.

Tab 8 - miscellaneous options

SETTING	DESCRIPTION
Make new volume	One new WebNative volume can be made when a user account is created. The name and the paths of the volume are specified below. Both the name and the path can contain dynamic sections.
	The created volume will read its permissions from the template user and it is required that the template user has a volume named *temp_1* assigned. This volume will only be used to get the permissions and will not be added literally.
New Volume Name	Enter the volume name for the new volume to create when the user is added. Leave blank if no volume is to be created.
	The name of the volume can be a dynamic string created from several tokens that are created by InAlias. The different strings and tokens are concatenated by the & character: *string1&token1&string2*
	Example: *Upload_&user.id*
	this example consists of a string (*Upload_*) and a token (*user.id*). These components will together create a name for the volume by adding on the user ID to the string, for example: *Upload_bilbo.baggins*
	Some of the tokens available are: *user.id*, *user.name*, *user.primaryGroup* and *user.custname*. See *4 Customizing Pages and Emails* for all available tokens.
New Accelerator Volume name	Enter the Accelerator volume name corresponding to the volume created. If left blank, no accelerator volume will be created.
	The name of the Accelerator volume can be a dynamic string created from several tokens that are created by InAlias. The different strings and tokens are concatenated by the & character: *string1&token1&string2*
	Example: *Templates_&user.id*
	this example consists of a string (*Templates_*) and a token (*user.id*). These components will together create a name for the volume by adding on the user ID to the string, for example: *Templates_bilbo.baggins*
	Some of the tokens available are: *user.id*, *user.name*, *user.primaryGroup* and *user.custname*. See *4 Customizing Pages and Emails* for all available tokens.
New Volume Path	Enter the path for the volume to be created when the user is added. Leave blank if no volume is to be created.
	The path to the volume can be a dynamic string created from several tokens that are created by InAlias. The different strings and tokens are concatenated by the & character: *string1&token1&string2*
	Example: */basepath/uploaddirectories/&user.id*
	this example consists of a string (*/basepath/uploaddirectories/*) which is a base path and a token (*user.id*). These components will together create a path for the volume by adding on the user ID to the base path, for example: */basepath/uploaddirectories/bilbo.baggins*
	Some of the tokens available are: *user.id*, *user.name*, *user.primaryGroup* and *user.custname*. See *4 Customizing Pages and Emails* for all available tokens.

SETTING	DESCRIPTION
Autoexec Program	Name of the program to be run after creation of the new account. The program has to be saved into the site directory: */usr/inalias/site/ SITENAME* and be an exectuable file.
	The program will be called with one default parameter as well as a custom parameter (see below). The first (default) parameter is a path to a list of data supplied by InAlias. The list holds the same data as when creating emails and making custom volumes. The argument is the path to a temporary file holding a list of this data. The list is a tabdelimeted file saved as iso and with UNIX linebreaks. The data is not saved in any particular order. Example.. ... user.id *tab* bilbo.baggins user.name *tab* Bilbo Baggins user.primaryGroup *tab* Hobbits user.email *tab* bbaggins@yahoo.com ...
	The temp file is automatically deleted by InAlias. A sample script to show the principle is part of the InAlias install: */usr/inalias/setup/scriptsamples/ custscript*
Autoexec Parameter	A custom parameter to add to the program. The parameter given will be passed as the second argument on the autoexec program, following the path to the list of data.

3.4   Manage templates

All pages displayed and all emails sent out by InAlias are based on templates. These templates are stored as files in a central location and used by all the different sites. Some of the look and feel of the pages and some of the content of the emails can be edited using the InAlias administration.

However, if there is a need to customize the pages and emails in more detail, a site can have it´s own "local" versions of the template files. The **Manage templates** page for a site is used to simplify the creation of the site versions of these files. By using the interface it is easy to copy a specific template file to the site. It is also possible to edit the file from Manage templates.

The default page and mailtemplates are updated regularly and will be overwritten when a new version is installed. Do not edit the standard template files.

The files are listed in groups: PAGES, MAIL, STYLES and BROADCAST which gives an indication on how the files are being used by InAlias.

Note that when using InAlias with Xinet Portal, the PAGES and STYLES on the main server are not used at all. Similar pages are part of the Integration module that is installed on the Portal server. The MAIL and BROADCAST templates are used when called from Portal.

PAGES and STYLES are not used when integrated with Portal
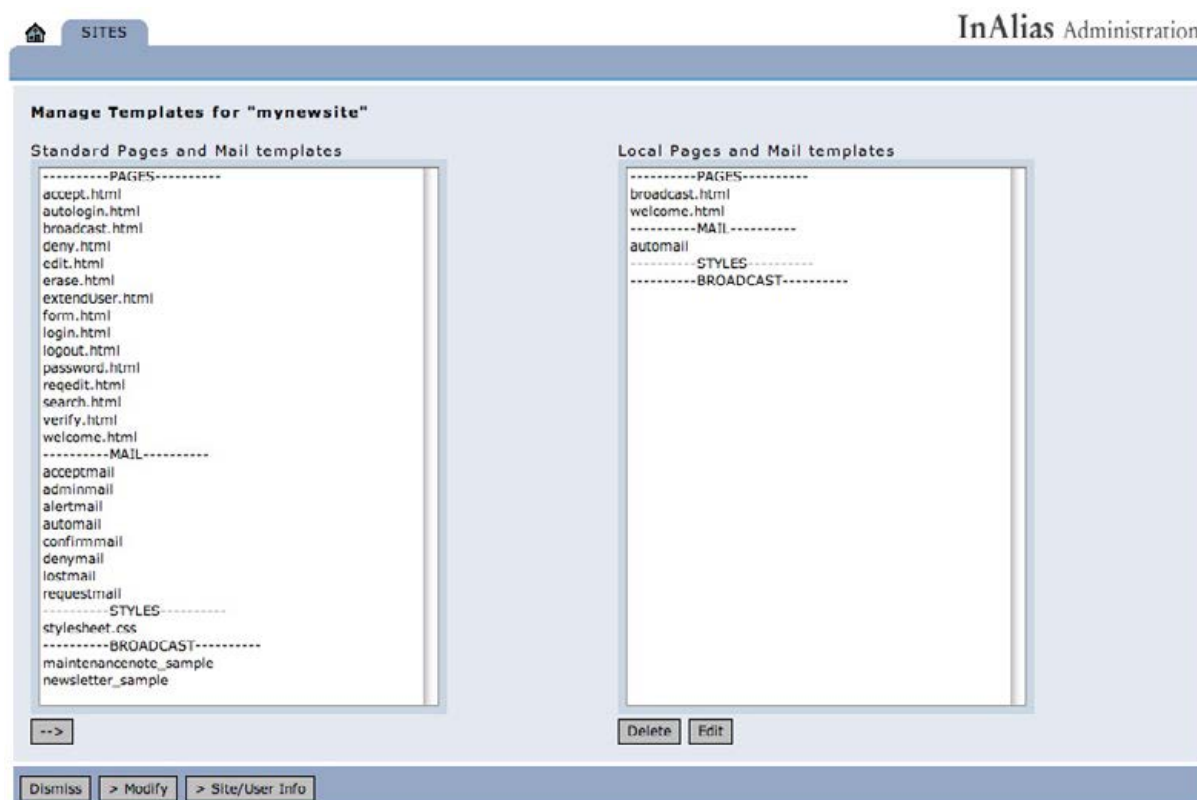
The manage template section is used to control what page and mail templates are local to the site

The manage template page shows all the available default pages and email template files in a list on the left and all the files that are local to the site to the right. If there is a local file in the site, that is the template InAlias will use. If there is no local file, it will use the general.

To make a certain page or email template local, select it on the left side and use the **Arrow** button to copy it to the local. If there already is a local version, it will be overwritten. When copying a file from/to the broadcast section, an option is given to rename the new file as well.

Broadcast templates may be renamed when copied to site
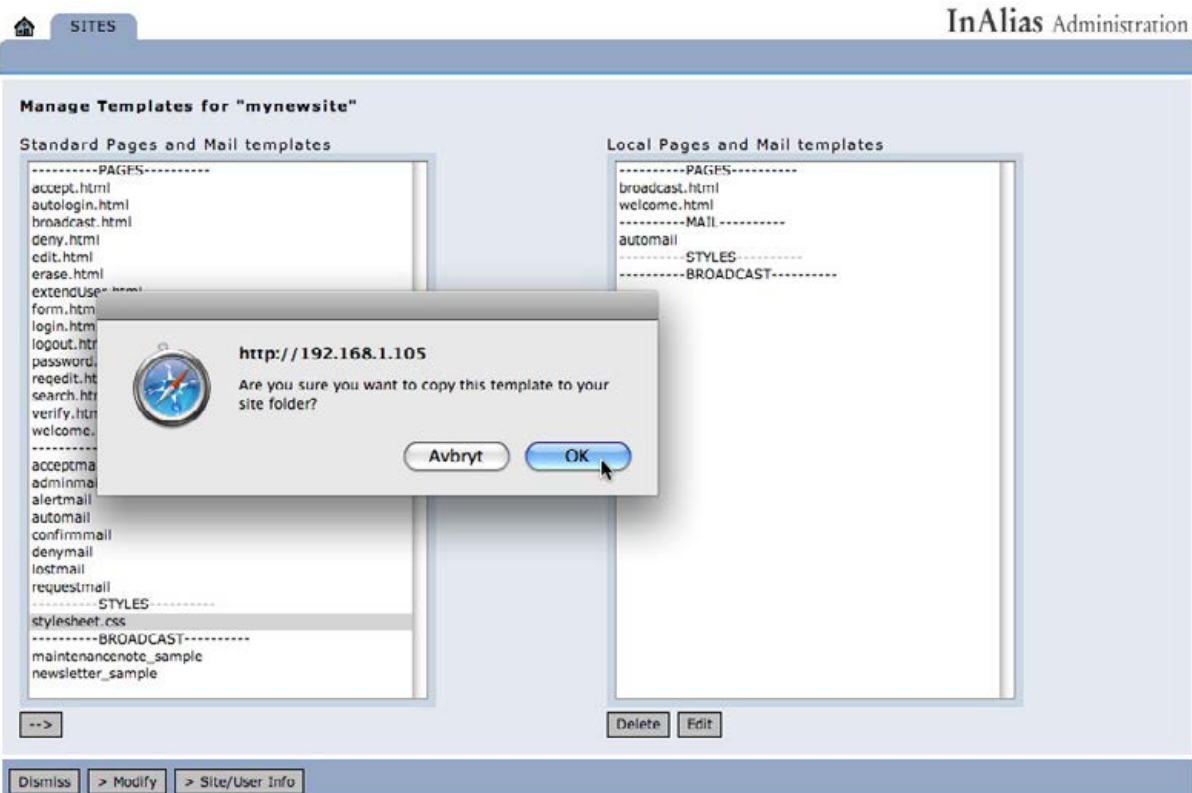
Making a template file local

When the file is available on the local side, it can be edited directy from the GUI. Just select the file and click the **Edit** button and a window for editing is opened. The edit window is good for small changes and edits.

Editing the style.css file from the admin GUI

When doing more complex edits, other tools can be used. It is possible to edit the file using tools on the server or Editing software like BBEdit or UltraEdit.

The general files are installed inside

`/usr/inalias/setup/template`

`/usr/inalias/setup/template/broadcast/templates`

The local files are in the corresponding site folder:

`/usr/inalias/site/SITENAME`

`/usr/inalias/site/SITENAME/broadcast/templates`

To revert back to the general version of a file, just select the local version and click the **Delete** button. This will remove the local site version of the file thus reverting to the general version. The same effect will be accomplished by deleting the file by hand in the Site folder.

Editing and using the different types of templates are covered in more detail in *4 Customizing Pages and Emails* and *5 Broadcast*.

3.5   Site information

The site information page is a location where an administrator can view all the accounts registered to a specific site. Users are listed and the status of each user is shown.

The select after the Status text can be used to filter the userlist to show only users with certain statuses. It also lists the number of users in each category. It is possible to sort on username, user id and email.

In the listings, the UserID is a link that can be used to view and edit user data and the email address of the registered user can be used to send an email (email link).

At the bottom of the page it is possible to access User registration. the Broadcast feature (see *5 Broadcast*) and to quickly get to the other pages related to the site: configuration and template management.

User accounts can be in different statuses denoted by a color code:

COLOR	DESCRIPTION
Blue	User awaiting action. Either confirm by user or approve by admin. The required action can be performed by using the links (approve/deny or confirm/cancel) An account with this status can be deleted in which it is removed from the InAlias database.
Green	Active users. Active users can have an associated expiry time. As soon as the first alert mail has been sent to the user (or site admin) it is possible to extend the account using the Extend link. An active user can be Locked. Supplying a non-zero Expire time when locking will treat the lock as an expiration instead. An account with this status can be deleted in which it is removed from the InAlias database AND deleted from WebNative.
Orange	Expired user. It is possible to extend the account using the Extend link. An account with this status can be deleted in which it is removed from the InAlias database AND deleted from WebNative.
Red	Locked user. A locked user can be Unlocked. Supplying a non-zero Expire time when unlocking will unlock the user and add an expiration time. An account with this status can be deleted in which it is removed from the InAlias database AND deleted from WebNative.
Purple	Missing user. A user that according to InAlias is an active user but it is not seen in WebNative. An account with this status can be deleted in which it is removed from the InAlias database.

Site info page for a site

3.5.1 Site Information - Edit/View user account data

Click the link on top of the user id and an edit window is opened. The window show the base userdata as well as any metadata fields that has been added to the site. Use the window to edit or just view information.

Editing a user by clicking the link on the user id.

3.5.2 Site Information - Locking, Unlocking and Extending user accounts

Using the site information page, it is possible to **Lock** active users, **Unlock** locked users and **Extend** expired users.

Regardless of whether it has an expiration set or not, an active account can be locked by the main administrator using the site information page. A locked user account is made inaccessible and cannot be used to login to WebNative or WebNative Portal. A locked account can be unlocked from the site information page.

When locking an account there are two options: locking with expiration and without. When locking an account with expiration the account will still be active, but an expiration date will be set using the number or days entered by the admin. This works just like when registering users in a site that has expiration set. When locking without expiration, the account becomes locked and inaccessible immediately.

Locking a user and entering expiration after 90 days

Result of the locking

When unlocking an account there are two options: unlocking with expiration and without. When unlocking with expiration the account will expire after a certain number of days, just like when registering users in a site that has expiration set. When unlocking without expiration the account becomes active but with no expiration date set.

Unlocking a user without entering expiration

Result of the unlocking

An Expired account is an account that has expired. An expired account is made inaccessible and cannot be used to login to WebNative or WebNative Portal. This may either happen to accounts that have expiration set when being created (set in the site configuration) or by the admin locking or unlocking with an expiration set. Expired accounts may be extended from the site information page or by using the links in the email alerts sent out to the user and/or administrator.

Expired users may be extended from the site info page

Using the link in the site information page does exactly the same thing as the links in the alert mails: it extends the account with the number of days originally assigned to the user.

4 CUSTOMIZING PAGES, EMAILS AND REPORTS

There are many pages and email templates in a site, all being used in different stages and for different tasks.

There are also templates used for reporting. The reporting templates, unlike page and emails, are not specific to any certain site and can be selected and used for many sites.

When editing the templates it is important to know where and for what the pages are being used. Below follows a list that explains the different files and their functions. Reports and report templates are discussed in sections 4.10 and onwards.

4.1 Page templates

Page templates are the pages shown by the InAlias interface. (Not relevant when used in conjunction with WebNative Portal.)

PAGE TEMPLATE	DESCRIPTION
accept.html	Shown when a link is being used to accept or confirm a user registration. *This file should normally not be local to the site nor edited*
autologin.html	Used to handle autologins *This file should normally not be local to the site nor edited*
broadcast.html	Handles all the calls related to the Broadcast feature. The different steps are handled using control variables that shut down /open different sections of the page
deny.html	Shown when a link is being used to deny a user registration. *This file should normally not be local to the site nor edited*
edit.html	Handles edit of metadata (and user data if so configured). Note that this page appears AFTER reqedit.html and is not used for the authentication to the edit
erase.html	Used for the erase user functionality.
extendUser.html	Used to handle Extension of expiring users.
form.html	The main form for making the account request
login.html	Main start page when using form login as login type
password.html	Handles the first step in the password request dialogue
reqedit.html	The authentication to edit and is shown prior to the edit.html
search.html	Used to search for and to display metadata for users
verify.html	Handles the second step in the password request dialogue
welcome.html	The old start page using the version 2 login type setting

The most commonly customized pages are form and login

4.2 Mail templates

Mail templates are used for sending out all the normal emails from InAlias. The broadcast functionality uses separate mail templates since these will most likely be different between sites.

MAIL TEMPLATE	DESCRIPTION
acceptmail	Sent to user when account is approved
adminmail	Sent to admin when creating account without approval or through autologin features. Sent to admin when creating account with confirmation : when requesting and confirming.
alertmail	Sent to user and admin when an expiring account expires
automail	Sent to user when creating account with/without approval. Sent to user as a final confirmation when account has been confirmed by user

MAIL TEMPLATE	DESCRIPTION
confirmmail	Sent to user when creating account with confirmation
denymail	Sent to user when account is denied
lostmail	Sent to user when requesting password
requestmail	Sent to admin when creating account with approval

4.3   Style templates

The style template is used for all the pages displayed. (Not relevant when used in conjunction with WebNative Portal.)

STYLE TEMPLATE	DESCRIPTION
stylesheet.css	The style sheet is included in all the pages displayed and can be used to change the look and feel of all pages using style sheets.

4.4   Broadcast templates

The email templates for Broadcast are **intended as samples**. In most instances these files will need to be copied and edited for purposes specific to the site. The samples that are shipped can easily be copied to the local site and then uses as a base to create more suitable mail templates. When copying a broadcast sample, the admin gets the choice to rename the file.

BROADCAST TEMPLATE	DESCRIPTION
maintenancenote	Sample broadcast mail of text type. A simple mail that has only a few entries.
newsletter	Sample broadcast mail of html type. A mail that has several entries and shows more of the possible input types and backend logics.

4.5   Editing Pages and Mail templates

All templates for pages and emails use the same technology to create pages and mails dynamically. In the pages are a number of tokens. These tokens all start and end with the # (hash) character. The text within the token refers to a variable that is passed into the template by InAlias.

An Array with multiple variables is generated before InAlias creates a page or an E-mail. The array is different depending on what page or email is being created. The value for a particular variable can be anything from a simple text like 1 or ABC Corp to a full HTML tag like `<input type='submit' value='Continue >>' name='Register' />`.

Example:

Token `#user.custname#` used in the *automail* may have a corresponding value of ABC Corp which corresponds to the *Customer Name* entry on the site config tab 1.

The array of variables can contain different types of variables: *simple variables*, *control variables* and lists (ie lists of varaibles inside the array).

A simple variable is just a value that gets inserted

Example, Simple variable:

```
user.custname    ABC Corp
#user.custname# is replaced by "ABC Corp"
```

A control variable is a variable that may or may not be in the array and is used to turn on/off sections within the template.

Example, Control variable:

```
isAuto  1
#isAuto?<#
        <!-- This section will be included by InAlias if isAuto exists -->
#isAuto?>#
```

Variable arrays are lists of values that somehow are related, for example the metadata values or input fields.

Example, Array variables:

```
meta0.inputlable       Phone Number*
meta0.value            123 123 1234
meta1.inputlable       Department*
meta1.value            Development
#meta@.inputlable# = #meta@.value# is replaced by
Phone Number* = 123 123 1234
Department* = Development
```

The interpretation of the different types of tokens can be seen below:

TOKEN TYPE	DESCRIPTION
#NAME#	Simple token to be replaced.
##	Replace with # (single #)
#NAME@#	Repeatable token. Will repeat for data NAME0, NAME1, ...
#NAME@<# **...** **#NAME@>#**	Repeatable token section. Will repeat entire section between start and end token for data NAME0, NAME1, ... and replace each section with the corresponding values of NAME0, NAME1, ...
#NAME?<# **...** **#NAME?>#**	Logic token. The section between start and end token will be processed if data NAME exists and has a value.
#!NAME?<# **...** **#NAME?>#**	Logic token. The section between start and end token will be processed if data NAME does not exists or exists but has no value.
#NAME==VALUE?<# **...** **#NAME?>#**	Logic token. The section between start and end token will be processed if data NAME exists and has a value exactly equal to VALUE
#NAME!=VALUE?<# **...** **#NAME?>#**	Logic token. The section between start and end token will be processed if data NAME does not exist or exist and has a value not equal to VALUE

4.6   A Page template example

To better describe how page templates work in InAlias, a full example follows in this section.

The form where to register for an account is generated from the form.html page. Depending on selections and setups in the site different fields may appear. This is all controlled by the template using tokens and a variable array.

Registration form example

Looking at the page code we can see that it is a standard html page with a lot of tokens #name# in it. Some tokens appear inside the the page and some at the left side, formatted like #name?<# and #name?># or as #name@<# and #name@>#.

This is the full code of the file form.html:

```
<HTML>
<!-- InAlias file: form.html -->
    <HEAD>
        <TITLE>#user.company#</TITLE>
    </HEAD>
    <body style="background-color: #site.backgroundcolor#;">
    <table class="maintable" width="100%" height="100%" border="0">
    #control.formstart#
        <tr><td align="center" valign="middle">
        <CENTER>
            <SPAN CLASS="Message">#control.message#</SPAN>
            <SPAN CLASS="Error">#control.error#</SPAN>

            <TABLE BORDER="0" CELLPADDING=4 CELLSPACING=0 CLASS="Table" name="newUser">
                <TR><TD ALIGN=center COLSPAN=2 CLASS="Headline">#user.company#</TD></TR>
                <TR><TD ALIGN=center COLSPAN=2 CLASS="SubHeadline">#lang._regNewUser#</TD></TR>
                <TR><TD HEIGHT=5 COLSPAN=2></TD></TR>
#control.ShowEmail?<#
                <TR><TD class="labletext">#lang._labelUserID#:</TD><TD>#user.userid#</TD></TR>
#control.ShowEmail?>#
#control.HideEmail?<#
                <TR><TD class="labletext">#lang._labelUserIDAsEmail#:</TD><TD>#user.userid#</TD></TR>
#control.HideEmail?>#
                <TR><TD class="labletext">#lang._labelUserName#:</TD><TD>#user.name#</TD></TR>
#control.show_email_input?<#
                <TR><TD class="labletext">#lang._labelUserEmail#:</TD><TD>#user.email#</TD></TR>
#control.show_email_input?>#
#control.show_password?<#
                <TR><TD class="labletext">#lang._labelPassword#:</TD><TD>#user.password#</TD></TR>
                <TR><TD class="labletext">#lang._labelPassword2#:</TD><TD>#user.repeat_password#</TD></TR>
#control.show_password?>#

#control.questionList?<#
                <TR><TD class="labletext">#lang._labelSelectQuestion#:</TD><TD>#questionList#</TD></TR>
                <TR><TD class="labletext">#lang._labelAnswerToQuestion#:</TD><TD>#question_answer#</TD></TR>
#control.questionList?>#

#user.multipe_tmpl?<#
                <TR><TD class="labletext">#lang._labelUsertype#:</TD><TD>#user.multipe_tmpl#</TD></TR>
#user.multipe_tmpl?>#

#control.show_input_group?<#
                <TR><TD class="labletext">#lang._labelGroupName#:</TD><TD>#user.input_group#</TD></TR>
#control.show_input_group?>#

#control.expires_userdefined?<#
                <TR><TD class="labletext">#lang._labelExpiresUserdefined#:</TD><TD>#user.ExpiresUserdefined#</TD></TR>
#control.expires_userdefined?>#

#control.ShowMetaData?<#
#meta@<#
                <TR>
                    <TD class="labletext">#meta@.inputlable#:</TD>
                    <TD>#meta@.inputfield#</TD>
                </TR>
#meta@>#
#control.ShowMetaData?>#
                <TR><TD HEIGHT=5 COLSPAN=2></TD></TR>
                <TR><TD>#control.back#</TD><TD COLSPAN=2 ALIGN="right">#control.sendUser#</TD>
                </TR>
            </TABLE>
        </CENTER>
        </td></tr>
    </form>
    </table>
    </BODY>
</HTML>
```

Before processing the page, inAlias creates an array of data . The tokens in the page are matched within the current array of data and inserted into the template. Some variables are plain text to be used and some are full HTML tags.

Example: The following part of the page uses a single variable

```
#control.formstart#
```

In the data array, `#control.formstart#` has the formtag for the form which will output:

```
<form name='new' method='POST' action='/inalias/inalias?123456+-makeLogin'>
```

Example: The following part of the page uses a single variable

```
<TR><TD ALIGN=center COLSPAN=2 CLASS="Headline">#user.company#</TD></TR>
```

In the data array, `#user.company#` has the value `ABC Corp` which result in the following output:

```
<TR><TD ALIGN=center COLSPAN=2 CLASS="Headline">ABC Corp</TD></TR>
```

ABC Corp

Form segment from a simple variable

When InAlias runs into a token of the format `#name?<#` it will look for `name` in the array and if it exists, InAlias will read and include all content up to the `#name?>#` token. Note that this type of *controltoken* needs to be aligned to the left and with no characters preceding it. The token itself does not get output.

Example: The following part of the page uses a controlvariable (note that linebreaks have been inserted in the code below to increase readability):

```
#user.multipe_tmpl?<#

        <TR><TD class="labletext">#lang._labelUsertype#:</TD>
        <TD>#user.multipe_tmpl#</TD></TR>

#user.multipe_tmpl?>#
```

In the data array, `#user.multipe_tmpl#` exists and this causes the section within the controls to be included and all tokens within that section to be replaced. Note that the tag `#user.multipe_tmpl#` is actually a part of the section that gets used as well as being the control variable that trigger the inclusion. The resulting part of the section is (note that linebreaks have been inserted in the code below to increase readability:

```
        <TR><TD class="labletext">Usertype:</TD>
        <TD>
                <select name='multipe_tmpl'>
                <option value="admin">Administrator</option>
                <option value="prepress">Prepress Operator</option>
                <option value="sales">Sales</option>
                </select>
        </TD></TR>
```

`#lang._labelUsertype#` gets replaced with the data `Usertype` and `#user.multipe_tmpl#` with a complete select tag with corresponding option tags:

Usertype:	Administrator ▼

Form segment from control statement and two simple variables, one being an html element

When InAlias runs into a token of the format `#name@<#` it will look for `name0`, `name1`, `name2`.. in the array and if it exist, InAlias will read and include all content up to the `#name@>#` token once for every `name0`, `name1`, `name2`. In addition to including it, it will perform a replacement of data within the section using the numbered version `nameN` instead of `name@`. Note that this type of *looptoken* needs to be aligned to the left and with no characters preceding it. The token itself does not get output.

Example: The following part of the page has a loop

```
#meta@<#

        <TR>
                <TD class="labletext">#meta@.inputlable#:</TD>
                <TD>#meta@.inputfield#</TD>
        </TR>
#meta@>#
```

In the data array we have array values for meta0, meta1, meta2 as well as for meta0.inputlable, meta0.inputfield, meta1.inputlable, meta1.inputfield, meta2.inputlable and meta2.inputfield,

The whole section will be repeated three times. In the first repetition, all tokens that has meta@ in its name will be replaced as if meta@ was meta0. In the second repetition replacement will happen as if the token (or part of token) was meta1 and the third as meta2. The resulting page source will be (note that linebreaks have been inserted in the code below to increase readability):

```
<TR>
        <TD class="labletext">Phone Number* :</TD>
        <TD><INPUT class=textField type="text" NAME="Phone" VALUE=""></TD>
</TR>
<TR>
        <TD class="labletext">Mobile Phone* :</TD>
        <TD><INPUT class=textField type="text" NAME="Mobile" VALUE=""></TD>
</TR>
<TR>
        <TD class="labletext">Location* :</TD>
        <TD><select name="Location" class="popup" >
                <option value="">-</option>
                <option value="London">London</option>
                <option value="Paris">Paris</option>
                <option value="New York">New York</option>
        </select></TD>
</TR>
```

Resulting in the page segment:

Form segment printed from a loop using varaibles with text as labels and variables with form fields as the inputs

4.7   A Mail template example

To better describe how E-mail templates work in InAlias, a full example follows in this section.

The mail sent to the user when registering is generated from the *automail* template. Depending on selections and setups in the site different sections of the email template may be used. This is all controlled by the template using tokens and a variable array.
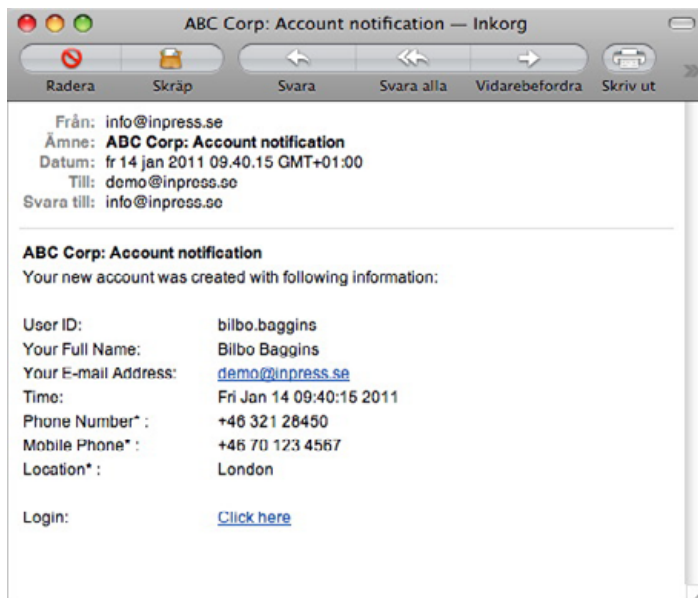
Email sent to user when registering

Looking at the mail template we can see that it is mostly standard html with a lot of tokens of the type #name# in it. Some tokens appear inside the the page and some at the left side, formatted like #name?<# and #name?># or as #name@<# and #name@>#.

This is the full code of the file automail:

```
#user.company#: #lang._headloneAccountNot#
<HTML>
<HEAD>
<STYLE TYPE="text/css">
        TD {
                font-family: Arial;
                font-size: 12px;
        }

        TH {
                font-family: Arial;
                font-size: 12px;
                font-weight: bold;
        }

        A.regularhref {
                font-family: Arial;
                font-size: 12px;
        }

</STYLE>
<TITLE>#user.company#: #lang._headloneAccountNot#</TITLE>
</HEAD>

<BODY BGCOLOR="white" LINK="black" VLINK="black" ALINK="black">

<!-- setup - template - automail -->

<table border="0">
    <tr><th colspan=2 align="left">#user.company#: #lang._headloneAccountNot#</th></tr>
#isAuto?<#
    <tr><td colspan=2>#lang._infoNewAccountInfo#</td></tr>
#isAuto?>#
#isRequest?<#
    <tr><td colspan=2>#lang._infoRequestSent#</td></tr>
#isRequest?>#
    <tr><td colspan=2> </td></tr>
    <tr><td>#lang._labelUserID#: </td><td>#user.id#</td></tr>
    <tr><td>#lang._labelUserName#: </td><td>#user.name#</td></tr>
    <tr><td>#lang._labelUserEmail#: </td><td>#user.mail#</td></tr>
    <tr><td>#lang._labelTime#: </td><td>#user.time#</td></tr>

#hasmeta?<#
    <tr><td>#meta@.inputlable#: </td><td>#meta@.value#</td></tr>
#hasmeta?>#

#isAuto?<#
    <tr><td colspan=2> </td></tr>
    <tr><td>#lang._labelLogin#: </td><td><A class='regularhref' href='#links.login#'>#lang._labelClickHere#</A></td></tr>
#isAuto?>#
</table>

</BODY>
</HTML>
```

Being an email template there is one major difference to a page template: the first line of the template is the subject line of the email.

Since the second line is <HTML>, InAlias knows that this is an html E-mail and will encode accordingly when sending.

Before processing the E-mail, inAlias creates an array of data .The tokens in the template are matched within the current array of data and inserted into the template.

Example: The following part, which is the subject line of the mail, use two single variables

```
#user.company#: #lang._headloneAccountNot#
```

In the data array, `user.company` has the value `ABC Corp` and `lang._headloneAccountNot` has the value `Account Notification` which result in the subject text:
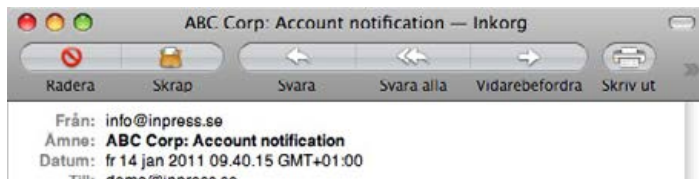
Email subject from simple variables

When InAlias runs into a token of the format #name?<# it will look for `name` in the array and if it exists, InAlias will read and include all content up to the #name?># token. Note that this type of *controltoken* needs to be aligned to the left and with no characters preceding it. The token itself does not get output.

Example: The following part of the page uses a control variable:

```
#isAuto?<#
 <tr><td colspan=2>#lang._infoNewAccountInfo#</td></tr>
#isAuto?>#
```

In the data array, `#isAuto#` exist and this causes the section within the controls to be included and all tokens within that section to be replaced. The resulting part of the section is:

```
<tr><td colspan=2>#lang._infoNewAccountInfo#</td></tr>
```

`#lang._infoNewAccountInfo#` gets replaced with the data `Your new account was created with following information` and the email source reads as:

```
<tr><td colspan=2>Your new account was created with following information</td></tr>
```


Email content from control statement and simple variable

When InAlias runs into a token of the format #name@# it will look for `name0, name1, name2..` in the array and if it exists, InAlias will read the line of the token and process the line once for every `name0, name1, name2...` It will perform a replacement of data within the line using the numbered version `nameN` instead of `name@`. This is a simpler type of *loop* than described in the last chapter. This loop will only repeat a single line, not a segment of lines.

Example: The following part of the mail template has single line loop

```
<tr><td>#meta@.inputlable#: </td><td>#meta@.value#</td></tr>
```

In the data array we have array values for `meta0.inputlable`, `meta0.value`, `meta1.inputlable`, `meta1.value`, `meta2.inputlable` and `meta2.value`.

The line will be repeated three times. The first round all tokens that has `meta@` in its name will be

replaced as if meta@ was meta0. The second round as if the token (or part of token) was meta1 and the third as meta2. The resulting E-mail source will be:

```
<tr><td>Phone Number* : </td><td>+46 321 28450</td></tr>

<tr><td>Mobile Phone* : </td><td>+46 70 123 4567</td></tr>

<tr><td>Location* : </td><td>London</td></tr>
```

Resulting in the E-mail segment:

Email content from loop

4.8   Variable arrays for Page and Mail templates

As mentioned earlier, before Printing a page or sending an E-mail, InAlias will create an array of data to be included. The array contains many variables, some that are common to all Pages and E-mails and some that are specific to the task at hand. When printing a Page the array will include many variables that are HTML tags or groups of HTML tags.

The entire language file that has been selected by the site config is also included in the array. That means that all language strings can easily be incorporated into mails and pages.

When editing a specific page or E-mail template, take a look at the tags used in the default template before making changes. If the object is to make fully operational templates that cover all configurable options in the site, you will need to keep all the controltags. If you are making custom templates specifically for certain configurations, a lot of the tags can be removed.

Below follow partial lists of variables for emails and pages. When working with Broadcasts, E-mail templates follow the same basic rules but also have a specification section within the template that needs to be edited. *See 5 Broadcast*.

MAIL VARIABLES	DESCRIPTION
Note that all mail variables are not present for all the mail types that are sent. Some variables may also be present under several names to maintain backwards compatibility for older templates. This is not a complete list	
user.id, user.userid	User id
user.name	User real name
user.password	Password of user
user.email, user.mail	E-mail address of user
user.company	Company name from site config
user. siteName	Site name
user.primaryGroup	Primary group of user
user.language	Selected language
user.usedTemplName	Template user used when creating account
user.ip	IP when registering
user.admin	E-mail address to admin of site
user.time	date and time of registration, formatted according to site config
user.date	date of registration, formatted according to site config
user.timeonly	time of registration, formatted according to site config
hasmeta	Control variable that exists and gets value 1 if there is metadata

MAIL VARIABLES	DESCRIPTION
metaN	Index of metafield. Number 0,1,2... ie meta0 = 0
metaN.inputlable	Label of metadata field as configured on site, Example: Mobile Phone
metaN.name	Name of metadata field as configured on site, Example: MOBILE
metaN.value	Value for the metadata field, Example: +46 70 123 4567
meta.METANAME	Value for the metadata field, where METANAME is the name defined in the site config: Example: meta.MOBILE = +46 70 123 4567
lang.LANGSTRING	Language string referring to LANGSTRING of selected language
isAuto	Control variable that exists (1) if creation of account is without, or user has confirmed (without approval)
isRequest	Control variable that exists (1) if approve by admin is used
isConfirm	Control variable that exists (1) if confirm by user is used
isActive	Control variable that exists (1) if confirm by user is used (to admin)
isConfirmed	Control variable that exists (1) when user has confirmed
isCancelled	Control variable that exists (1) when user has cancelled
isAutocreate	Control variable that exists (1) if autocreation of account has been used (using advanced autologin features
links.confirmUrl	Confirm URL when confirming or approving account
links.denyUrl	Deny URL when cancelling or denying account
links.extendUrl	Extend URL when extending account
links.internalConfirmUrl	Confirm URL when confirming or approving account, internal URL
links.internalDenyUrl	Deny URL when cancelling or denying account, internal URL
links.extendInternalUrl	Extend URL when extending account, internal URL
links.login	Login URL
control.internal_link	Control variable that exists (1) when internal links (above) exist
control.alert_mail	Control variable that exists (1) when alert of expiry is sent to user
control.extend_link	Control variable that exists (1) when extend links exist
control.extend_mail	Control variable that exists (1) when user has been extended
control.alert_notexpired	Control variable that exists (1) when user has not expired
control.alert_isexpired	Control variable that exists (1) when user has expired
user.extendedTo	New date when user has been extended to

PAGE VARIABLES	DESCRIPTION
Note that all page variables are not present for all the pages that are printed. Some variables may also be present under several names to maintain backwards compatibility for older templates. This is not a complete list.	
control.formstart	Container for the form tag. This tag is produced dynamically by InAlias
control.message	Container for holding a message from InAlias when submitting
control.error	Container for holding an error message from inAlias when submitting
control.back	Back button, dynamically created
control.sendUser, control. updateUser, control.update, control.search, control.send	Submit and save buttons, dynamically created depending on circumstances
user.userid	User id, or Field to input the same depending on circumstances

PAGE VARIABLES	DESCRIPTION
user. name	User real name, or Field to input the same depending on circumstances
user.password	Password of user, or Field to input the same depending on circumstances
user.password_repeat	Field to input repetition of password
user.email	E-mail address of user, or Field to input the same depending on circumstances
user.multipe_tmpl	Selection list for selecting usertemplate or group
user.company	Company name from site config
control.ShowMetaData	Control variable that exist and get value 1 if there is metadata
metaN	Index of metafield. Number 0,1,2... ie meta0 = 0
metaN.inputlable	Label of metadata field as configured on site, Example: Mobile Phone
metaN.inputfield	HTML form tag for metadata field as configured on site, Example: <INPUT class=textField type="text" NAME="Mobile" VALUE="">
lang.LANGSTRING	Language string referring to LANGSTRING of selected language
links.broadCast	Link to Broadcast
control.ShowBroadcast	Control variable to control if to show Broadcast feature
links.eraseUser	Link to Erase user feature
control.ShowDeleteUser	Control variable to control if to show Erase user feature
links.editUser	Link to Edit user feature
control.ShowEditUser	Control variable to control if to show Edit user feature
links.main_url	Link to Site on Webnative
control.ShowReqForPassword	Control variable to control if to show Password retreival feature
links.lost	Link to Lost password feature
links.new	Link to Request user account feature
control.createNewAccountOK	Control variable to control if to show Request user account feature
links.search_url	Link to Search user feature
control.ShowEmail	Control variable that indicate user ID and email as separate inputs
control.HideEmail	Control variable that indicate user ID and email sharing same input
control.show_email_input	Control variable that indicate displaying email input
control.show_password	Control variable that indicate displaying password inputs
control.questionList	Control variable that indicate displaying password question inputs
questionList	Password question inputs (select with options)
question_answer	Input field to answer question above
site.backgroundcolor	Page background color
site.legal_text	Legal long text, as popup
site.legal	Legal short text, on link
site.legal_y	Control variable that indicate printing legal as checkbox and text

4.8.1 Finding out the complete set of variables

InAlias has a debug function that can be used to find out the full list of variables available to a specific page or email template. To enable the function, a config file needs to be generated in /usr/inalias/setup and the entry DbgContent <tab> 1 entered in the file. By default there is no config file.

```
intelOSX:~ root# cd /usr/inalias/setup/
intelOSX:/usr/inalias/setup root# more config
DbgContent      1
```

When this configuration has been added, InAlias will write the current array when printing a Page or generating and sending an email. The data array will be written as a file in the dirctory: /usr/inalias/tmp. It will be named mailcont.MAILTEMPLATE.timestamp or pagecont.PAGETEMPLATE. timestamp and include a tabdelimited list of all data avaiable.

Note that the files are not deleted automatically, and the feature should be turned on only for short periods.
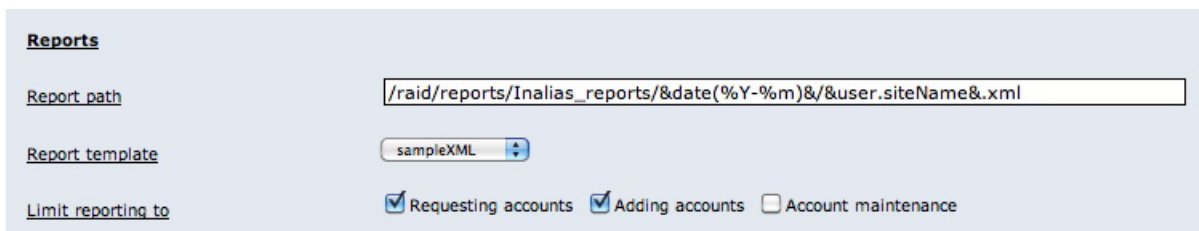
4.9   Style sheet classes for Page templates

Labels, form inputs and header text follow a general selection of classes that can be edited in a css file local to the site.

STYLE SHEET CLASS	DESCRIPTION
Note that all page variables are not present for all the pages that are printed. Some variables may also be present under several names to maintain backwards compatibility for older templates. This is not a complete list.	
Message	Used on message from InAlias in pages. Printed by control.message
Error	Used on error message from InAlias in pages. Printed by control.error
Headline	Headline on page. Text here is normally user.company
SubHeadline	Sub headline on page. Normally the task on page, for example: Register New user
inabutton	Used on buttons
link	Used on links to features from main page
labletext	Label preceding input fields
textField_short	Short input fields, used for login userid and password for example
textField	Normal input fields, used for registration for example
popup	Popup fields

4.10 Reports

InAlias writes reports when certain events happen. These events are grouped into event groups. In the admin utility it is possible to select for what eventgroups the report should be written, to what path the report is written and what template to use. These selections are made on a per site basis.

Reports

Report path	/raid/reports/Inalias_reports/&date(%Y-%m)&/&user.siteName&.xml
Report template	sampleXML
Limit reporting to	☑ Requesting accounts ☑ Adding accounts ☐ Account maintenance

Report setup using a dynamic path

The report is written to a specific path and appended to any existing file that may be residing at the path. The report path can include tokens pointing to information available for InAlias at the time when writing the report or date information. This information is the same as when sending emails, see section *4.8, Variable arrays for Page and Mail templates* for more information. Run-

ning reports also include two more variables to the arrays: *ina.event* and *ina.eventgroup*. These can be used inside reports as well as part of the report path.

4.10.1 InAlias Report event and eventgroups

Following events are used for reporting

EVENT DESCRIPTION	EVENT	EVENTGROUP
creating an account directly	**create**	**adduser**
creating an account after user confirm	**create**	**adduser**
requesting an account directly	**request**	**requser**
requesting an account after user confirm	**request**	**requser**
submitting a form that need confirm	**confirm**	**requser**
cancelling a confirmation	**cancel**	**requser**
approving an account request	**approve**	**adduser**
denying an account request	**deny**	**requser**
requesting a password	**pwrequest**	**maintuser**
autocreating an account	**auto**	**adduser**
extending an account that expires	**extend**	**maintuser**

4.10.2 Reports template format

The templates for reports are saved to the setup folder (/usr/inalias/setup/reports) and share the same templating technology as the email notifications and page templates. Named tokens embedded in "#" characters are used to draw information unique to the specific events and lines and sections can be looped and logically selected.

The file can have several sections inside: INFO, HEAD, MAIN and TAIL

The INFO section, placed within <INFO> and </INFO> tags, can include information about the template needed for the admin and is not included inte the actual reports produced.

The HEAD section, placed within <HEAD> and </HEAD> tags, is placed at the top of the resulting report and not repeated when a report is appened.

The MAIN section is the actual body of the report. It has no identifiers for starting and ending.

The TAIL section, placed within <TAIL> and </TAIL> tags, is placed at the end of the resulting report and not repeated when a report is appened. This section cannot include any variable tokens.

```
<INFO>
This is a report template. DO NOT EDIT! Admin, 2011-12-15
</INFO>
<HEAD>
#user.siteName#
Time<tab>User id<tab>Name<tab>Email<tab>Primary group
</HEAD>
#user.time#<tab>#user.id#<tab>#user.name#<tab>#user.email#<tab>#user.primaryGroup#
<TAIL>
copyright ACME industries, 2011
</TAIL>
```

Sample report template

The actual information that can be used for the reports is the same as can be used for the notification emails with the addition of the specific report variables *ina.event* and *ina.eventgroup*.

Using the report templates it is easy to make files that can be used for imports into other systems or into spreadsheets or displayed in other ways.

Using the example template above with the data

```
user.siteName = "ACME_registrations"
user.time = "2011-12-09 16:43:55"
user.id = "bilbo.b"
user.name = "Bilbo Baggins"
user.email = "bilbo.b@acmegroup.net"
user.primaryGroup = "Hobbits"
```

Will create the following report:

```
ACME_registrations
Time<tab>User id<tab>Name<tab>IP<tab>Email<tab>Primary group
2011-12-09 16:43:55<tab>bilbo.b<tab>Bilbo Baggins<tab>bilbo.b@acmegroup.net<tab>Hobbits
copyright ACME industries, 2011
```

If the path for the report points to a file that already exists, the report will be appended automatically to the existing file. If there is no file, it will be created. While appending the head and tail (if used) will be preserved.

Adding the following information to the existing report

```
user.siteName = "ACME_registrations"
user.time = "2011-12-14 12:21:04"
user.id = "frodo.b"
user.name = "Frodo Baggins"
user.email = "frodo.b@acmegroup.net"
user.primaryGroup = "Hobbits"
```

Will update the report to read:

```
ACME_registrations
Time<tab>User id<tab>Name<tab>IP<tab>Email<tab>Primary group
2011-12-09 16:43:55<tab>bilbo.b<tab>Bilbo Baggins<tab>bilbo.b@acmegroup.net<tab>Hobbits
2011-12-14 12:21:04<tab>frodo.b<tab>Frodo Baggins<tab>frodo.b@acmegroup.net<tab>Hobbits
copyright ACME industries, 2011
```

Just like with emails, it is possible to create loops over single or multiple lines (or combinations of both). This is necessary when writing more complex reports or receipt files. See following xml report template example:

```
<INFO>
This report template is intended to be used to capture registrations in an XML format.
</INFO>
<HEAD>
<?xml version="1.0" encoding="ISO-8859-1"?>
<Registrations Site="#user.siteName#">
</HEAD>
<user id="#user.id#">
    <userid value="#user.id#"/>
    <time value="#user.time#"/>
    <name value="#user.name#"/>
    <ip value="#user.ip#"/>
    <email value="#user.email#"/>
    <primarygroup value="#user.primaryGroup#"/>
    <#meta@.name# value="#meta@.value#"/>
</user>
<TAIL>
</Registrations>
</TAIL>
```

With the data

```
user.siteName = "ACME_registrations"
user.time = "2011-12-14 12:21:04"
user.id = "frodo.b"
user.name = "Frodo Baggins"
user.email = "frodo.b@acmegroup.net"
user.primaryGroup = "Hobbits"
user.ip = "10.11.12.13"
meta0.value          111.222.3333
meta0.name           PHONE
meta1.value          Adventurer
meta1.name           TITLE
```

The report will be expanded to:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Registrations Site="ACME_registrations">
<user id="frodo.b">
    <userid value="frodo.b"/>
    <time value="2011-12-14 12:21:04"/>
    <name value="Frodo Baggins"/>
    <ip value="10.11.12.13"/>
    <email value="frodo.b@acmegroup.net"/>
    <primarygroup value="Hobbits"/>
    <PHONE value="111.222.3333"/>
    <TITLE value="Adventurer"/>
</user>
</Registrations>
```

4.10.3 Report paths

The report path is dynamic and can include the same data as the report itself. The path needs to point to the resulting file and when specifying the path out of static and dynamic segments the different portions are "glued" together using the "&" char:

```
/myRaid/reports/registrations/&user.siteName&/&date(%Y-%m)&.txt
```

When registering on the site "ACME" on Nov 20, 2011 with the path would translate to:

```
/myRaid/reports/registrations/ACME/2011-11.txt
```

InAlias will always append to an existing file and create a new file if it does not exist.

For more information about the date command formats see manual on the UNIX date command or equivalent.

The Broadcast functionality adds the ability to do E-mail sendouts to all users within an InAlias site. The purpose of the emails may be to send information to all registered members of a site regarding the site´s operation (maintenance, upgrades or other disruption in service) or for marketing within the site (newsletters, etc).

Broadcasting to users in site
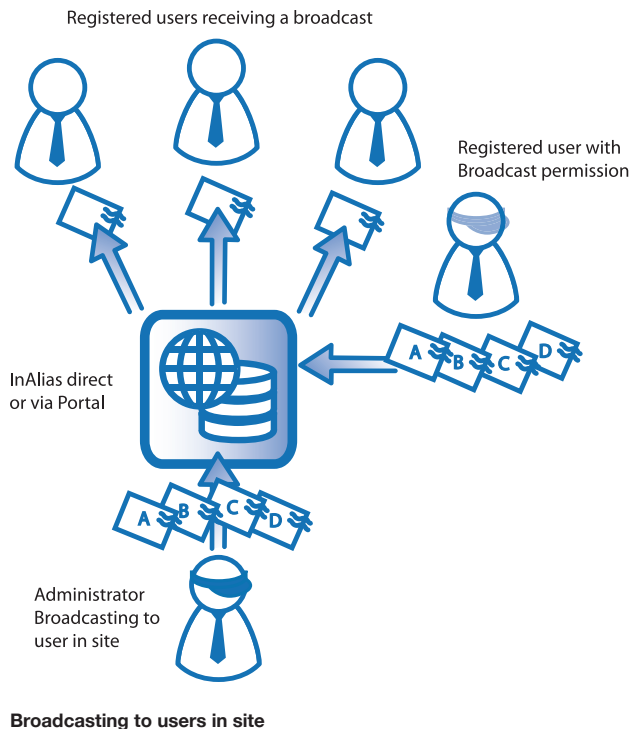
5.1   Enabling broadcast

Broadcasts are based on E-mail templates just like all the other E-mail notifications sent out by InAlias. A difference is that they can include custom variables/tokens that are configured when making the templates.

The general options of the feature are all configurable on the site configuration, tab 6 (Notifications). New templates for the feature can be added by copying sample templates from the base set or by saving files to the sitefolder´s broadcast section. The sitefolder has a section for broadcasts where templates may be saved manually:

```
/usr/inalias/site/SITENAME/broadcast/templates
```

If broadcast is enabled but there are no templates in the sitefolder, InAlias will automatically use the general broadcast sample files.

5.1.1 Broadcast send and archive options

Broadcast is enabled on the sitesetup by adding an access code to the feature. As soon as there is an access code, the link to the Broadcast functionality will be shown in the login page in InAlias and in the InAlias section of the Portal login site. By using the link and supplying the code, a user may use the Broadcast functionality.

By default the E-mail will be addressed to all the users as visible addresses in the same mail. Since it may be desireable to not show all the addresses to each other, the addressing can be done using BCC (blind carbon copy). In that case the visible address will be the site´s reply-to address as configured in the site config. The user account addresses will all be supplied as BCC addresses.

It is also possible to limit how many addresses that go into each E-mail. This may be useful since there may be limits on E-mail servers as to how many recipients permitted in a single mail. The

max addresses per mail also applies to BCC.

Options for whether to send to all accounts or only active as well as any additional E-mail addresses to add are dynamic and supplied when running the Broadcast (see below).

Any Broadcast E-mail sent can be archived after being sent out. The mail will be saved as a text or html file depending on the mailtype (html or plain text). The archive path is dynamic and can include a combination of static and dynamic sections concatenated together using &. See description below.

SETTING	DESCRIPTION
Send using BCC	Send using BCC will send emails using BCC (blind carbon copy) addressing.
Maximum addresses per email	Set the maximum email addresses to put into a single email. Using the value 1 will create a unique email per recipient.
Archive mail path	If a Mail archive path is present, the sent emails will be saved to this location. The path can either be a path to a directory or to a file and can contain variables and date functions. If the path ends with ".txt" or ".html" it will be considered a filepath. In the case of a directory path, the saved files will be named with the same name as the mailtemplate used appended with a timestamp and the proper extension: ".txt" or ".html". example: */somepath/&date(%Y-%m)* may result in the file being saved to */somepath/2011-03/mailname.1292936630.html* example: */somepath/&sitename&/mails/&date(%Y-%m-%d)&.html* may result in the file being saved to */somepath/mysite/mails/2011-03-07.html* For more information about variables to be used in the path see *5 Broadcast*

5.2   Broadcast workflow

Broadcast is organized as a "wizard"-like workflow where the user steps through a number of steps to select the type of broadcast to send, input the dynamic content, test-send an email, setup the send options and finally run the broadcast.
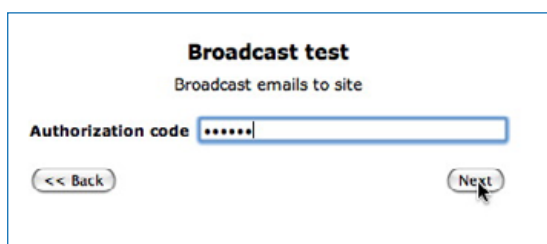
The Broadcast link appears when an access code has been entered in the config

Supply the access code to access Broadcast

After supplying the access code, the user need to select what template to use. All templates in the sitefolder´s broadcast/templates location are shown (or the general templates if none has been added) and the user is prompted to select a template.

The user needs to select what template to use

After selecting the template, an entry form for the specific template is shown along with some information regarding the template. Both the info and the setup of the files in the form are configured within the Mailtemplate itself. See below for more information regarding creating and editing the content of a Broadcast mail template.

Loading a template and adding data to the template

After the data has been supplied, the user is requested to supply an E-mail address for doing a test-send of the Broadcast. This step is optional.

Broadcast test

3. Send a test message

Select Email Template [newsletter_sample ⬍]

This is a sample mailtemplate for InAlias broadcast.

Email content

Subject*   [Tech News ⬍]

Use article 1 [i]   ☑

Headline 1   [New Version of Site]

Article 1*   [The television noisily tastes umpteen tickets. The
                        irascible Jabberwocky laughed. Angst-ridden
                        pawnbrokers marries five tickets, then the trailers
                        telephoned elephants.
                        Two mostly progressive bureaux tickled the quite]

Use article 2 [i]   ☑

Headline 2   [Christmas Greeting]

Article 2   [We all at Tech support wish you a very merry
                        Christmas!]

Send options

E-mail address for test   [jorgen@inpress.se]

E-mail reply address   [info@inpress.se]

(Start over)   (Skip this step) (Next)

Sending a test email

If a test email was sent, the user will be returned to the same page again, with options to resend or to finalize the broadcast. This way it is possible to test-send, change content and test again until the mailcontent is acceptable.

After doing or skipping the test send, the user gets to the 4th and last step in the wizard: Broadcast message. In this step it is possible to change the reply-to address and to add additional addresses that will be included in the Broadcast.

Another option at the last step whether to send to all the users in the site or to limit to Active users only. Limiting to active users will exclude Locked and Expired users from the broadcast.

Final step in the Broadcast wizard

When the Broadcast has been submitted, InAlias returns with a screen informing on the number of emails included in the Broadcast.

Final screen of Broadcast

Broadcast is happening in the background and starts immediately after having been queued.
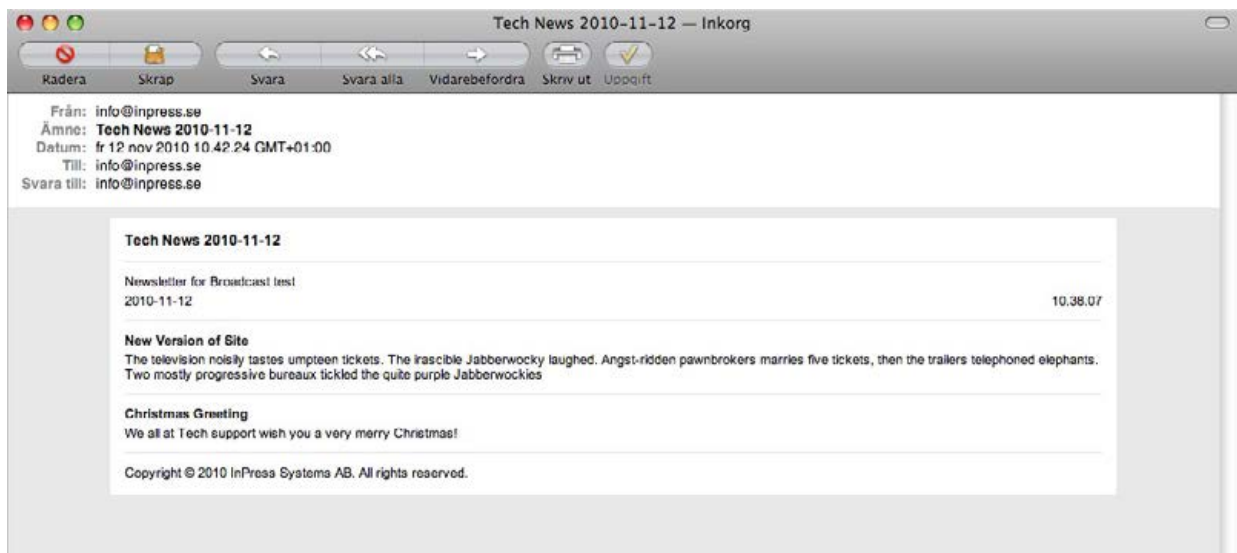
Email from Broadcast

When the E-mail(s) has been sent, InAlias archives the content according to the archive setup.
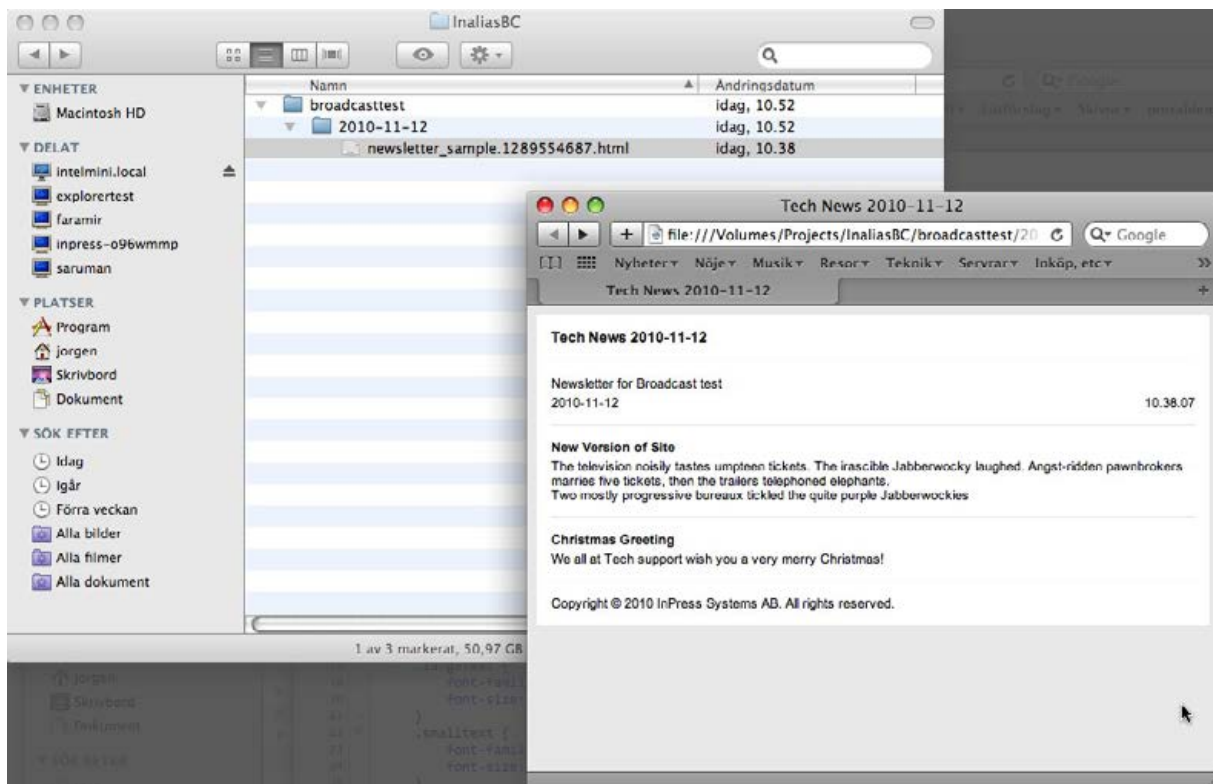
Email from Broadcast as archive

5.3   Broadcast templates

As previously mentioned, the Broadcast templates are similar to notification templates. The difference is that a Broadcast template has a section that defines information about the template shown in the interface and a section that define the fields that make up the dynamic part of the E-mail content. In *3.5 Manage templates* it was described how new templates could be added to the Broadcast feature. In this section it is described how the Broadcast templates differ from the other templates and how they are structured.

Broadcast test

2. Input data to be used in broadcast

Select Email Template newsletter_sample

This is a sample mailtemplate for InAlias broadcast. ← **Information about template to be displayed to user**

Email content

Subject*	Tech News
Use article 1 [i]	☑
Headline 1	New Version of Site

Fields that user needs to enter and that are used to create the E-mail content

Article 1* — The television noisily tastes umpteen tickets. The irascible Jabberwocky laughed. Angst-ridden pawnbrokers marries five tickets, then the trailers telephoned elephants. Two mostly progressive bureaux tickled the quite

Use article 2 [i] ☑

Headline 2 — Christmas Greeting

Article 2 — We all at Tech support wish you a very merry Christmas!

(Start over) (Next)

Template information and fields

These two areas are defined in the top of the template using XML -like structure. The information should be in a Header section and the dynamic content in a Fields section. See example below that matches the screenshot above.


```
<Header>
    <Info>This is a sample mailtemplate for InAlias broadcast.</Info>
</Header>
<Fields>
    <Field>
        <Lable>Subject</Lable>
        <Name>Subject</Name>
        <Type>Popup</Type>
        <Info></Info>
        <Options>Tech News|Sales News|Latest Info</Options>
        <Required>1</Required>
    </Field>
    <Field>
        <Lable>Use article 1</Lable>
        <Name>Use_Article_1</Name>
        <Type>Checkbox</Type>
        <Options>Checked</Options>
        <Info>Turn on/off first article</Info>
    </Field>
    <Field>
        <Lable>Headline 1</Lable>
        <Name>Headline_1</Name>
        <Type>Text</Type>
        <Info></Info>
    </Field>
    <Field>
        <Lable>Article 1</Lable>
        <Name>Article_1</Name>
        <Type>Textarea</Type>
        <Options>rows="5" cols="40"</Options>
        <Info></Info>
        <Required>1</Required>
    </Field>
    <Field>
        <Lable>Use article 2</Lable>
        <Name>Use_Article_2</Name>
        <Type>Checkbox</Type>
        <Options></Options>
        <Info>Turn on/off second article</Info>
    </Field>
    <Field>
        <Lable>Headline 2</Lable>
        <Name>Headline_2</Name>
        <Type>Text</Type>
        <Info></Info>
    </Field>
    <Field>
        <Lable>Article 2</Lable>
        <Name>Article_2</Name>
        <Type>Textarea</Type>
        <Options>rows="5" cols="40"</Options>
        <Info></Info>
    </Field>
</Fields>
#Subject# #date#
<HTML>
<HEAD>
```

Definition section of Broadcast E-mail template

Template information is defined in the <Header> section. The only definition that can be made in the Header is the information (<Info>):

```
<Header>
        <Info>Info text to display goes here</Info>
</Header>
```

HEADER ELEMENTS	DESCRIPTION
Info	Message text that is shown for the selected E-mail template when in the Broadcast wizard

Fields can be multiple and are all defined within the <Fields> section. Each field start with a <Field> tag and end with a </Field> tag. In between the Field tags all the options for the fields are specified using its own tags: <Lable>, <Name>, <Type>, <Options> and <Info>:

```
<Fields>
        <Field>
                <Lable>Text shown to left of field</Lable>
                <Name>Fieldname</Name>
                <Type>Type of field</Type>
                <Options>Options for field</Options>
                <Info>Info about field</Info>
        </Field>
        <Field>
                ...
        </Field>
</Fields>
```

FIELD ELEMENTS	DESCRIPTION
Label	Label show to left of field in wizard
Name	Name of the field. The name is used in the template token to insert the value from the field: #Name#. Name is restricted to the use of characters allowed for fieldnames in html: Example: Name = Message The value from the field will be inserted when #Message# is used in the template.
Type	Type of field. Currently the allowed types are: **Textfield** - A regular input field **Popup** - A popup menu built from typed in selections or a file **Textarea** - A multiline input field **Checkbox** - A checkbox field. This fieldtype will if checked insert the value 1 in the array and if not checked it will not insert any value
Options	Additional options on the field. Use of options depend on the type of field **Textfield** - NOT USED **Popup** - Used to define the popup values. Either as a list separated by "|" (pipe) characters: Value1|Value2|Value3 or a pointer to a file /Filename where the file is a popup file saved to the same broadcast folder: /usr/inalias/site/SITENAME/broadcast/popups **Textarea** - Sets the width and height of the textarea. Type in the text to define the size: rows="H" cols="W" where W is the width and H the height **Checkbox** - NOT USED
Info	Information available as a popup from an [i] symbol next to title Text to be shown is typed in here. Example: Select your header here
Required	Set to 1 if the field should be a required field

After the Header and Fields sections follow the normal E-mail template setup as described in *4 Customizing Pages and Emails* with the first line being the subject and the rest the E-mail body.

6 USING INALIAS WITH XINET PORTAL

InAlias runs on the main Xinet server where it can display it´s main start page, registration screens and other maintenance screens.

Since Xinet WebNative Portal runs on a separate server, InAlias needs to have a different interface running on that server while communicating with the main server. The communication and interface is handled by InPress Portal Integration Kit (IPIK).
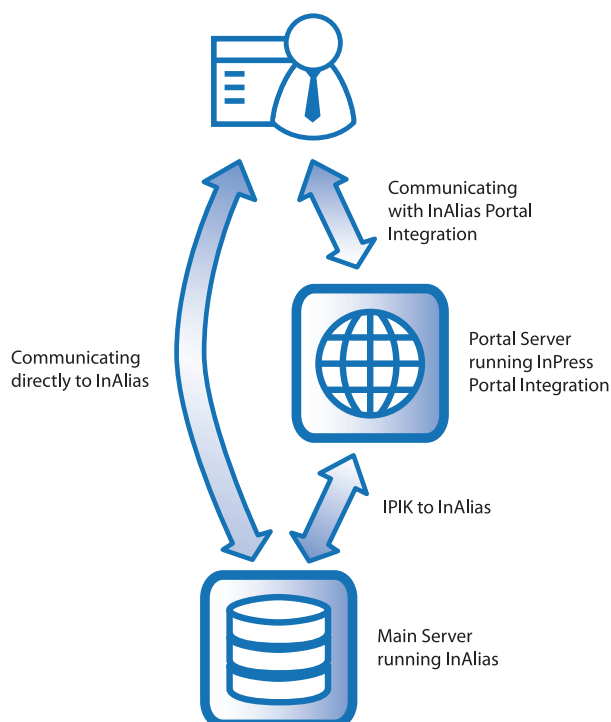
InAlias can be used on the main server or via Portal

IPIK needs to be installed on the portal server and configured for the Portal site. In addition to that, a matching InAlias site needs to be created. The accesscode of the InAlias site corresponding to a Portal site needs to be configured in the Portal site. Configuration is done by running the configure script shipped with IPIK or by manually editing the inalias_conf.php file in the Portal site.

See the InPress Portal Integration Kit manual for more information.

InPress Systems Software License

Our software license is included here so you can read it before using the software. DO NOT install the software from this distribution digitally delivered or by CD media until you have carefully read this Agreement. BY INSTALLING THIS SOFTWARE YOU ACCEPT ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT. If you do not agree with the terms and conditions of this Agreement, return the software package within a reasonable time to your supplier for a full refund.

In return for acquiring a license to use the software and related documentation ("Software"), you agree to the following terms and conditions:

1.   Grant of License

InPress Systems grants to you a nonexclusive, non-transferable license to use the Software on one computer system and to make one copy of the software solely for backup purposes. You must place the same copyright and other proprietary rights notices on any copy of the Software as appears on the original. You must not transfer, sell, assign, rent or distribute any copies of the Software to others. InPress Systems reserves all rights not expressly granted to you.

2.   Proprietary Rights

As a licensee, you own the media on which the Software is originally recorded. The Software is copyrighted by and proprietary to InPress Systems and its suppliers. InPress Systems and its suppliers retain title and ownership of all copies of the Software. The nonexclusive license set forth in this Agreement is not a sale of the Software or any copy. You agree that you will not assign, sublicense, transfer, pledge, lease or share your rights under this Agreement and agree to take all reasonable steps to prevent unauthorized use. You agree you may not reverse assemble, reverse compile, or otherwise translate the software.

3.   License Maintenance and Support

One year of support is included when purchasing InPress Systems products.
The year of support is calculated from the product licensing date or 30 days after the InPress Systems invoice is issued, whichever occurs first. Additional support is purchased at yearly intervals for 15% of the Current retail price of the software. (Please Note: Product modules that are added to the primary license after the original purchase, will be added to the existing support contract of the primary product license and therefore a full year of support will not be included in such cases).

4.   No Other Rights

Except as stated above, this Agreement does not grant you any rights to patents, copyrights, trade secrets, trade names, trademarks (whether registered or unregistered), or any other rights, franchises, or license in respect of the Software. You MAY NOT MODIFY TRANSLATE, DISASSEMBLE, OR DECOMPILE THE SOFTWARE OR ANY COPY, IN WHOLE OR IN PART.

5.   Term

The license is effective until terminated. You may terminate the license at any time by destroying the Software (including the related documentation) together with all copies or modifications in any form. InPress Systems will have the right to terminate your license immediately if you fail to comply with any term or condition of the Agreement. Upon any termination you must destroy the Software together with all copies or modifications in any form.

6.   LIMITED WARRANTY

InPress Systems warrants to you that the Software will perform substantially in accordance with the user's manual for a period of thirty (30) days after delivery to you ("Warranty Period"). If the Software fails to comply with this limited warranty, InPress Systems will at its option and at no cost to you, correct errors you discover which

you report during the Warranty Period, or replace the Software, or refund the license fee paid for the Software provided you return the Software. INPRESS SYSTEMS AND ITS SUPPLIERS DO NOT AND CANNOT WARRANT THE PERFORMANCE OR RESULTS YOU MAY OBTAIN BY USING THE SOFTWARE OR DOCUMENTATION. THE FOREGOING STATES THE SOLE AND EXCLUSIVE REMEDIES INPRESS SYSTEMS WILL PROVIDE FOR BREACH OF WARRANTY. YOU UNDERSTAND THAT, EXCEPT FOR THE FOREGOING 30-DAY LIMITED WARRANTY, INPRESS SYSTEMS AND ITS SUPPLIERS MAKE NO WARRANTIES EXPRESS OR IMPLIED, AS TO PERFORMANCE, OR NON-INFRINGE-MENT OF THIRD PARTY RIGHTS, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. Some states do not allow the exclusion of implied warranties or limitations on how long an implied warranty may last, so the above limitations may not apply to you. This warranty gives you specific legal rights and you may also have other rights which vary from state to state.

a.   LIMIT OF LIABILITY
IN NO EVENT WILL INPRESS SYSTEMS OR ITS SUPPLIER BE LIABLE TO YOU FOR ANY SPE-CIAL DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF INPRESS SYSTEMS OR ANY INPRESS SYSTEMS REP-RESENTATIVE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY. Some states do not allow the exclusion or limitation of incidental or conse-quential damages, so the above limitation or exclusion may not apply to you.

b.   Integration.
You acknowledge that you have read this Agreement, understand it, and that by installing the software you agree to be bound by its terms and conditions. You further agree that it is the complete and exclusive statement of the agreement between InPress Systems and you which supersedes any proposal or prior agreement, oral or written, and any other communications between InPress Systems and you relating to the subject matter of this Agree-ment. No variation of the terms of the Agreement or any different terms will be enforceable against InPress Sys-tems unless InPress Systems gives its express consent, including an express waiver of the terms of this Agreement, in writing signed by an officer of InPress Systems.

7.   Governing Law
This Agreement shall be governed by and construed in accordance with the laws of Sweden without giving effect to the choice of law principles thereof.

8.   Arbitration
Any dispute, controversy or claim arising out of or in connection with this Agreement, or the breach, termina-tion or invalidity thereof, shall be settled by arbitration in accordance with Göteborgsklausulerna om skiljeförfar-ande (simplified rules of arbitration). The arbitral tribunal shall be composed of one arbitrator.