

# Advanced website customization using templates

[Reasons for using custom templates](#)

[How FirstClass web templates work](#)

[Server-side includes](#)

[Creating templates](#)

[Coding conventions](#)

[Where to put custom templates](#)

[How templates work with FirstClass objects](#)

[Form objects](#)

[Listings objects](#)

[Exceptions](#)

[How template tags work](#)

[Message tag examples](#)

[Common standard templates](#)

[Container lists](#)

[Subcontainer lists](#)

[Leaf item lists](#)

[Combined item lists](#)

[Directory lists](#)

[History lists](#)

[Search lists](#)

[Who's Online lists](#)

[Calendar lists](#)

[Instant Messaging](#)

[Calendar events and tasks](#)

[Messages](#)

## Reasons for using custom templates

If you want to make basic changes to the FirstClass web interface, such as different fonts and colors, you can use the .sitepref form. More elaborate changes, such as custom forms, require you to customize the templates that ship with FirstClass.

If you support multiple websites on your system, you can create unique templates to reflect this diversity. For example, if you have different sites for both your Sales and Engineering departments, you could create template sets that reflect the different perspectives of these departments.

Before you begin, you should be familiar with:

- FirstClass Internet Services
- your operating system
- basic HTML and a solid understanding of Internet Services script variables.



Note

Advanced information, such as script variables, can be found on FCOL at Conferences > Peer to Peer Support > FirstClass Webmasters > FAQs.

[Top](#)

## How FirstClass web templates work

All Internet Services web templates and [server-parsed files](#) incorporate the components of IS script, which is an extension of Apache's Extended Server-Side Include scripting language. These components are:

- X-FC tags

Pull data from the FirstClass Network Store (FCNS) on the server and display FirstClass content to the web user.

- commands  
Provide some flow control when data is being processed and displayed to the web.
- IS Script functions  
String processing functions (can be part of any IS Script expression).
- IS Script variables.  
Used in any expression, static value, or function argument by prefixing the variable name with a \$ sign.

Client forms have corresponding templates for rendering the forms to the web. Like client forms, web templates retrieve data from the data store on the FirstClass server through form and field ID numbers. Embedded X-FC tags act as placeholders for fields on the template. These X-FC tags map directly to fields on a client form and pull the same data from the FirstClass server. Each field on the client form corresponds to the same field on the web template.

## Server-side includes

Server-side includes are instructions to the server to include a specified file in the current document before sending the document to the user who requested it (similar to a library item that is provided by the server). Server-side `#include` syntax in a document inserts a reference to an external file; it does not insert the contents of the specified file in the current document.

When someone opens a document, the server processes the `#include` instructions and creates a new document. In the new document, the `#include` instructions are replaced by the contents of the included file. The server then sends this new document to the browser. This processing takes more time to serve out to the web than other pages.

To edit the contents of a server-side include file, the file must be edited directly. Any changes to the included file are automatically reflected in every document that contains it.

[Top](#)

---

## Creating templates

Customizing templates is an advanced task that requires a good working knowledge of the FirstClass client, FirstClass IS script, HTML, HTTP protocol, advanced Javascript, and Cascading Style Sheets (CSS). You also need to understand the structure and components of web templates at the tag level.

You can create custom templates in different ways:

- copy an existing template into a FirstClass document or text editor, and customize the code
- create a new document in FirstClass or a text editor using HTML, JavaScript, and Internet Services script.

If you create a template in an HTML editor, and you want the template to be editable, you must manually create both the read only and editable templates. Regardless of how you create a web template, it must consist of a noneditable (read only) version and an editable version. Noneditable templates are named *formid* and editable templates are named *Edit.formid*.

Editable templates are usually larger and more complex than their corresponding read only counterparts, due to the extra embedded code to initiate specific actions. The graphic below shows a section of the read only and editable Message templates. The syntax circled in **orange** identifies the template on the right as editable:

**read-only message template (141)**

```

<!--141 T=<X-FC-REQUEST-TIME>--><!--#if expr="<X-FC-BROWSER
NAME.MSIE>&&<X-FC-BROWSER OS.Mac>"--><!--#set var="ALTSTR"
value="title"--><!--#else--><!--#set var="ALTSTR" value="alt"--><!--#endif--><!--#set
var="PRINT" value="YES"-->
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=<X-FC-ITEM
DATA[Charset]>">
<title><X-FC-ITEM DATA[Subject]></title>
<style type="text/css">
<!--
<!--#include virtual="/templates/GlobalStyles.inc"-->
-->
</style>
</head>
<!-- T=<X-FC-REQUEST-TIME> -->
<body bgcolor="<X-FC-FIELD PREFS.5600 DEFAULT=LANG.49.5600.11 RGB>"
leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">
<form action="<X-FC-OBJURL SLASH>FormEdit"><!--#set var="NavPosition"
value="<X-FC-FIELD PREFS.5513 DEFAULT=LANG.49.5513.11 NUMBER>"--><!--#if

```

**editable message template (Edit.141)**

```

<!--E141 T=<X-FC-REQUEST-TIME>--><!--#if expr="<X-FC-BROWSER NAME.MSIE>&&
<X-FC-BROWSER OS.Mac>"--><!--#set var="ALTSTR"
value="title"--><!--#else--><!--#set var="ALTSTR" value="alt"--><!--#endif--><!--#set
var="PRINT" value="NO"-->
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=<X-FC-ITEM
DATA[Charset]>">
<title><X-FC-ITEM DATA[Subject]></title>
<style type="text/css">
<!--
<!--#include virtual="/templates/GlobalStyles.inc"-->
#body{width:100%;}
-->
</style>
<script language="JavaScript">
<!--#include virtual="/templates/JavaScript/SendFunctions.inc"-->
</script>
</head>
<body bgcolor="<X-FC-FIELD PREFS.5600 DEFAULT=LANG.49.5600.11 RGB>"
leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">
<form name="MainForm" action="<X-FC-OBJURL SLASH>Send" method="post"
enctype="multipart/form-data" onSubmit="if(CS){CS=0;}else{return false;}">
<input type="hidden" name="CharSet" value="<X-FC-ITEM DATA[Charset]>">
<input type="hidden" name="KeepNames" value="1"><!--#set var="NavPosition"
value="<X-FC-FIELD PREFS.5513 DEFAULT=LANG.49.5513.11 NUMBER>"--><!--#if

```

Here are some examples of editable and noneditable templates:

- a Message template: 141 and Edit.141

When you create a New Message on the web, you are using the Edit.141 (editable) template. When your recipient reads the message, she is opening the 141 (noneditable) template.

- a Phone Call template: 128 and Edit.128

When you create a New Phone Call on the web to send a Called No message notification, you are using the Edit.128 (editable) template. When your recipient reads the phone message, she is opening the 128 (noneditable) template.

- a Personal Address template: 119 and Edit.119.

When you create a New Personal Address on the web, you are using the Edit.119 (editable) template. When you open this address on the web, you are opening the 119 (noneditable) template.

Look in your templates folders for other examples.

When a web user creates a new object (for example, a message or a document), the first thing Internet Services does is look for and retrieve the matching template form ID as its corresponding client form. If Internet Services cannot find a template that matches the form ID, it will load template 95 as the default. If Internet Services cannot find an editable template that matches the form ID of the object, it will load the read only version of that template.

In noneditable templates (read only), you can only view the document displayed, not edit any information. For example, when you receive an email message and open it, you can only read it. To edit the message, you must either reply to the message or forward it on to another user. When you perform this action, the displayed template becomes editable, since you are actually opening up the editable version of that same template.

In all cases, use Internet Services script variables to customize your templates. Internet Services script includes:

- X-FC tags
- global constants
- commands
- functions.

Internet Services script works only in server-parsed files. Internet Services, as shipped, considers the following files to be server-parsed:

- files ending in .shtml, .shtm, .swml
- include files (.inc)
- templates (located in the Template Sets folder).

- forms created with FirstClass Designer and saved as an HTML document.

In the MIME Types file, you can define whether Internet Services should server parse a file type.

## Coding conventions

The shipping web templates use:

- **blue** for all X-FC tags, to differentiate them from static code or other Internet Services script syntax
- **magenta** for embedded values in an X-FC tag  
For example, `<X-FC-FIELD `1000 + $INDEX` LONG>.`
- **red** for server-side include statements  
For example, `<!--#include virtual="/.templates/GlobalStyles.inc"--><!--#set var="COLOUR" value=".`
- **black** for HTML or Javascript code  
For example, `<meta http-equiv="Content-Type" content="text/html; charset=<X-FC-CHARSET">".`
- **green** for comments.  
For example, `<!--#rem General Purpose Globals:...-->.`

This color coding convention is just for convenience, to improve the readability of the templates. Although not necessary, we recommend you follow this convention, as it greatly simplifies template debugging.



Note

X-FC tags and arguments are case insensitive.

## Where to put custom templates

If you want your custom templates to be the default web interface, put them in the current template set folder, by default called Standard Templates.

When you are arranging your templates in the WWW folder, make sure the templates are in the correct folders and that your Multiple Sites & Languages form is correctly configured to reflect your site setup.

[Top](#)

---

## How templates work with FirstClass objects

To customize web templates, you need to understand the different levels of FirstClass objects and how you build templates to view them. Objects are grouped into three categories:

- form objects
- listings objects
- exceptions.

### Form objects

Form objects are forms that reside on the server and contain data. Form objects can be:

- mail forms (read only and editable versions)
- documents (read only and editable versions)
- memforms (edit only)
- system forms (read only and editable versions)
- calendar forms (read only).

### Mail forms

Mail forms specifically support the Send operation. Mail forms include:

- messages
- phone call forms
- file transfer forms.

## Documents

Documents are items that are saved to the server but not sent (do not support the Send operation).

## Memforms

Memforms are virtual forms that do not represent objects on the server but behave as if they do. Memforms mimic documents, because they are saved but not sent. These forms are associated with actions and include:

- Auto Registration
- Change Password.

## System forms

You cannot create or delete system forms. System forms include:

- Preferences
- System Profile.

## Calendar forms

Calendars are considered forms when they are viewed by month or by week instead of as a list.

## Listings objects

Listings objects are displayed as lists (for example, message listings in your Mailbox or document listings in a conference). All listings are composed of multiple templates and have IDs less than zero.

Listings objects consist of:

- [container](#) listings and [leaf objects](#)
- search listings
- Directory listings
- Who's Online listings
- history listings.

## Container listings and leaf objects

Containers hold and display listings of other subcontainers and leaf objects. Examples of container objects are:

- conferences
- folders
- calendars.

Examples of leaf objects are:

- messages
- documents
- bookmarks
- events
- tasks
- uploaded files.

## Filtering container listings

The output of container and leaf objects in a listing can be limited to a specific range (typically used to introduce 'paging' to reduce download time). Both container and leaf objects each maintain separate ranges, which can be controlled at the URL parameter level by using nItems, nFolders, FirstItem, FirstFolder, LastItem and LastFolder. If you wish to see the deleted items in a container listing, use the 'Show deleted' URL parameter (for example, "<X-FC-OBJURL>?Showdeleted = 1").



Note

If you use more than two tags or items, the third will be ignored. Internet Services only reads the first two tags or items in either the object URL or the template code.

## Sorting and grouping of container listings

Sorting and grouping sorts a list by whatever criteria you specify in the URL parameters. By default, your sorting parameters follow those set in the client. If you want to change the sorting and grouping behavior, you do so using URL parameters. The URL parameter syntax is:

```
[I|F]table[commit]=SortCol[_GroupCol]
```

Where,

- `SortCol` changes the sort of the columns
- `GroupCol` changes the grouping of the columns.

The parameters for both `SortCol` and `GroupCol` are `([-] ColID +1)`. The `(ColID + 1)` is a numeric expression that is evaluated. If `SortCol < 0`, this implies Reverse sort. If `GroupCol < 0`, this implies collapse groups.

Where,

- `[commit]` saves the new sort or group column to the server (it commits your changes)
- `[I|F]` sorts on Item or Folders (make sure that you have a split in your view before using the `[I]` literal term).

For a list of current valid columns, see the X-FC-LAYOUT tag.

## Container default listing mode

A default listing for a container object is a combination of different templates. It is divided into five sections and follows this order:

- 1 Header templates (-1 and -10)
- 2 Subcontainer listing templates (-11, -12, -14)
- 3 Separator template (-15)
- 4 Leaf listing templates (-16 to -18)
- 5 Footer template (-2)

If you don't want all of the contents of a container listed, you can abort the process at any point by including the `<X-FC-HEADER-ONLY>` tag in the last template you want listed.

## Container component listing mode

Internet Services provides the functionality to filter listings based on the types of objects it contains (for example, subcontainers or listings). You can obtain these listings by adding a parameter to the container URL. Let's take a look at the different URL parameters required:

- to filter for subcontainers only, use the 'Folders' URL parameter.
- to filter for leaf only items, use the 'Items' URL parameter.
- for a combined listing, use both parameters.

All parameters use the -205, -206, and -207 templates.

Here is an example that works for all three cases:

```
[<X-FC-OBJURL>?Folders = 1-10]
```

This URL lists the first ten subcontainers in a container.



Note

The filtering argument can also be used as a paging argument using the syntax: `filter=first-last`. For more information, see the Filtering section in this document.

## Search listings

The Search form is a system form that allows you to specify your search filter criteria in FirstClass. The form ID for this system form is 8000. When you do a search (using the 8000 system form) your results are posted using these templates:

- -300 (Search Listing Header)
- -301 (Search Listing Item)

- -302 (Search Listing Progress)  
This template renders the changed search folder.
- -303 (Search Listing Footer).

## Directory listings

The FirstClass Directory is a list of everything in the system to which you can address mail (for example users, conferences, address books, and aliases). Directory listings consist of:

- Directory filters
- Directory search listings
- name validation
- Who's Online.

## Directory filters

Directory filter parameters are used to display only a portion of the Directory. This is useful, as there are usually a large number of items contained in the Directory and the user only wants to view a sublist.

When a user performs a search, FirstClass filters either on Type or Name. When the filter is on Type, you can filter the Directory entries for:

- local names (users on local system plus all address book entries)
- remote names (users on gateway systems)
- conferences (all conferences whose Directory listing is visible to users)
- address book (all address book entries)
- Internet alias (all Internet address aliases)

All these filters are integers and use 1 for 'on' and 0 for 'off' in the URL parameter. The default filter is local/remote/conference.

When the filter is on Name, you can search on a full name or part of a name. In either case use the 'srch' parameter. Also, if you are using a browser, it is a good idea to set your 'charset' parameter to the correct character set of your system (ISO character set codes).

## Directory search listings

To open a simple Directory listing (for example, send mail to a user, open a user calendar, or open a profile) use the <X-FC-LOGIN>Directory URL. The templates used for this type of listing are:

- - 100 (Directory form)
- - 104 (Directory form footer).

To do a Directory lookup, use the Directory search listings URL <X-FC-LOGIN>Directory/Lookup. The templates used for this type of listing are:

- -100 (Directory form)
- -101 (Directory list header)
- -102 (multiple results require multiple -102 templates) or -105 (no results)
- -103 (Directory list footer).

## Name validation

To validate names in a field, use the <X-FC-OBJURL SLASH>Lookup URL, where <X-FC-OBJURL > is in whatever object you are validating the names into (where the server puts the names once they are validated).

Some objects (for example, resource or location calendars) have limited Directory listings so filters are used. The templates used for this type of listing are:

- -100 (Directory Form)
- -101 (Directory List Header)
- -102 (one for each name request)
- -105 (only if there are name requests)
- -103 (Directory List Footer).

Name validation sometimes supports and, in some cases, requires these extra parameters:

- Operation
- Insert|Insert Multi

This parameter is used to search for a recipient to insert in a recipient list. The Insert|Insert Multi parameter uses these parameters:

- Field ID = fieldid

This is a mandatory parameter that specifies the field into which you are inserting.

- [Index = index]

This specifies where in the list you are going to insert the recipient name.

- [standard Directory parameters]

These parameters are standard Directory parameters used to filter the Directory.

If the Insert parameter returns a single match, the recipient name is inserted directly into the object on the server. For example, if you open a message and type in an exact match for a name in the To field, the name is automatically added to the list in the field. There is no need for a Directory listing. The Insert Multi parameter uses the same syntax as the Insert parameter but inserts multiple names into the object on the server.

- Delete

The Delete parameter is used to delete a name from a recipient list and its valid values are:

- Field ID = fieldid

This parameter specifies the field that you are deleting from.

- [Index = index][client ID = clientid]

This parameter must have either an index parameter (one at a time) that deletes a name as per its position in the list, or one or more client ID parameters (removes first instance of the client ID).

### Example of Index parameter

If your To list has name1, name2, name3, this URL would delete name1:

```
<X-FC-OBJURL SLASH>Lookup?Operation=Deleted&FieldID=9&Index=0>
```

- Add

The Add parameter is used to add a name from a recipient list and its valid values are:

- Field ID = fieldid

This parameter specifies what field you are adding to.

- [Index = index]

This parameter specifies where in the list to place the new names.

- Client ID = clientid

This parameter must have one or more client IDs. If you have specified an Index, the client IDs are added in reverse order. If no Index is specified, no order is defined.

### Who's Online listings

A Who's Online listing is a list of all users currently logged in who you are allowed to see. To open a Who's Online listing use the URL "<X-FC-LOGIN> whosonline". Templates used for this listing are:

- -400 (Who's Online Header)
- -401 (Who's Online Item)
- -402 (Who's Online Footer).

### History listings

A History listing lists the history of any leaf objects (for example, a calendar event, message, or document). To open a history listing use the URL "<X-FC-OBJURL SLASH>history".

The templates used are:

- -110 (History Header)
- -111 (History Item)



- -112 (History Listing Footer).

## Exceptions

Exceptions are neither objects nor listings. They include stationery, instant messaging, and monitors. In a filtered container list, however, these exceptions are listed within subcontainers. Internet Services does not inherently render these objects to the web.

[Top](#)

## How template tags work

A web template is coded according to what FirstClass content it must display using X-FC tags and other IS script variables. Depending on the nature of the tag, X-FC tags pull data directly from one of two places:

- the en.rez resources file (mainly guide text)
- the server.

Data pulled from the server comes from a number of places:

- the current object (for example, the "To" or "Subject" fields of a message or the list of items in a conference)
- the session data, which is the information about the server or the current user (for example, the user's full name, the server site name, or the server time zone, and so on)
- preferences (for example, any data located on the .sitepref form and the user Preferences form).

Generally, the most common X-FC tags used in FirstClass web templates fall into three categories:

- X-FC FIELD
- X-FC ITEM (and corresponding tags, such as, X-FC-LIST-ITEM)
- X-FC BODY.

## Message tag examples



### Caution

Randomly pasting any of the following code examples into a template may not work as desired. Do not consider any of the code shown in these examples as complete.

client form

A screenshot of a client form. It features a 'Subject' label followed by a text input field. To the right of the input field is a small box containing the number '7'. The input field itself has a small box containing the number '6'.

web form

A screenshot of a web form. It shows the label 'Subject:' followed by a text input field.

- `<X-FC-FIELD FORM.6.9 STRING>`

Maps to the "Subject" field text, where FORM means the current form, 6 is the "Subject" field ID, 9 is the property ID (label/guide text), and STRING designates the output of this tag as a string.

- `<X-FC-ITEM DATA[Subject]>`

This tag represents the actual "Subject" user-input field.

client form

A screenshot of a client form. It features a 'Cc:' label followed by a text input field. To the right of the input field is a small box containing the number '11'. The input field itself has a small box containing the number '10'.

web form

A screenshot of a web form. It shows the label 'Cc:' followed by a text input field and an 'Add' button.

- `<X-FC-FIELD FORM.10.9 STRING>`

Maps to the "Cc" field text, where FORM means the current form, 10 is the "Cc" field ID, 9 is the property ID (label/guide text), and STRING designates the output of this tag as a string.

- `<X-FC-RECIPIENT-COUNT CC>`

Outputs the number of recipients in the CC list.

- `<X-FC-FIELD LANG.7528.13 STRING>`

Outputs the *lang.rez* string resource #7528, index 13 (in this case "Delete Recipient").

- `<X-FC-RECIPIENT CC $RECIPIENTNUM DATA[Name]>`

Extracts the name of the \$RECIPIENTNUM th CC name.

- `<X-FC-RECIPIENT CC $RECIPIENTNUM DATA[Name] DIRNAME>`

Extracts the name of the \$RECIPIENTNUM th CC name. DIRNAME applies bolding, italics, and other formatting to the recipient based on whether or not they are online, able to accept Instant Messaging, and so on.

client form



web form



- `<X-FC-FIELD FORM.12.9 STRING>`

Maps to the "Attachments" field text, where FORM means the current form, 12 is the "Attachments" field ID, 9 is the property ID (label/guide text), and STRING designates the output of this tag as a string.

- `<X-FC-LIST-COUNT>`

Outputs the # of (unhidden) attachments for this message.

- `<X-FC-LIST-ITEM-URI ` $ATTACHMENTNUM` BASE>`

Outputs the URL to the \$ATTACHMENTNUM th attachment. The BASE argument removes the file name from uploaded file attachments, which is necessary when doing operations (for example, deleting them).

- `<X-FC-LIST-ITEM ` $ATTACHMENTNUM` DATA[Subject] LENGTH>`

Outputs the length of the subject (in bytes) of the \$ATTACHMENTNUM th attachment in the list.

- `<X-FC-LIST-ITEM ` $ATTACHMENTNUM` DATA[Subject]>`

Outputs the subject of the \$ATTACHMENTNUM th attachment in the list.

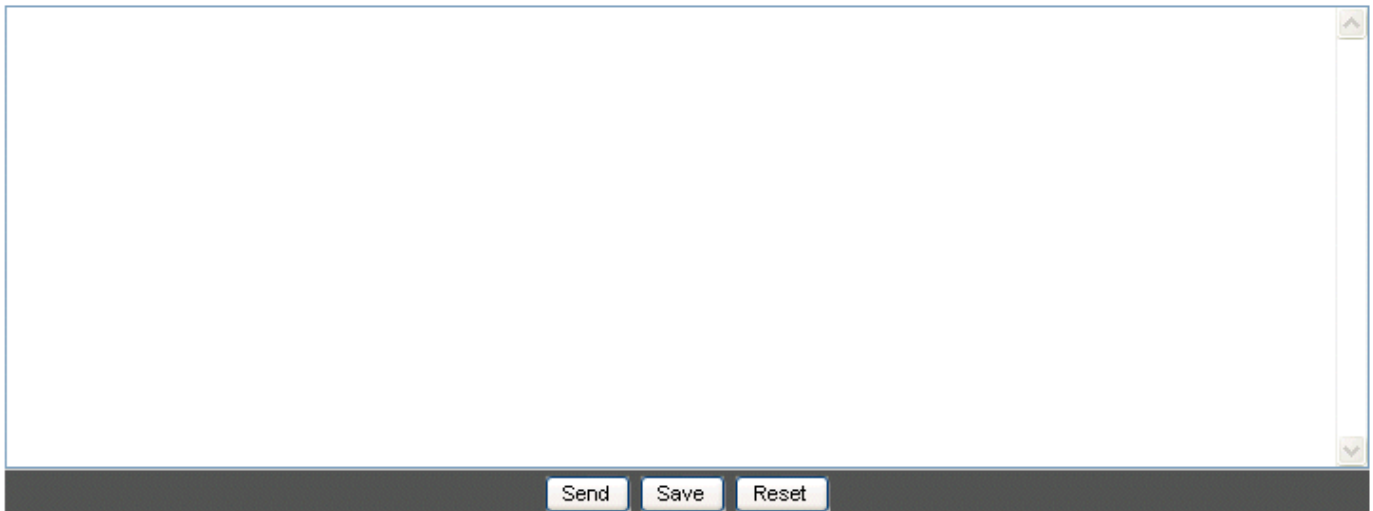
- `<X-FC-LIST-ITEM ` $ATTACHMENTNUM` DATA[Size] SCALED>`

Outputs the size of the \$ATTACHMENTNUM th attachment in the list. The SCALED argument tells Internet Services to use the appropriate symbol given the size of the attachment (for example, K, M, G, T).

client form



web form



- **<X-FC-BODY>**

This tag represents the actual "Body" user-input field.

[Top](#)

---

## Common standard templates

These are commonly used FirstClass templates that may appear in our template sets.

Depending on the purpose of a particular template set, you may not require some of these templates. In any template set, there will usually be additional templates for specialty forms and other functions.

## Container lists

These templates output the whole container listing with no filters. This means that if you use these templates, both subcontainers in the main container and leaf objects are shown.

- -1 the container list master header
- -2 the container list master footer
- -10 the container list master header
- -11 the subcontainer list header
- -12 the subcontainer list row separator (outputs every four subcontainers)
- -13 the folder list item (outputs one for every subcontainer)
- -14 the folder list footer
- -15 the list separator between subcontainers and leaf objects
- -16 the leaf list header
- -17 the leaf list item (outputs one for every leaf item)
- -18 the leaf list footer (if there is one).

## Subcontainer lists

These templates output container folder listings with certain filters. This means that if you use these templates, only folder container objects are listed.

These templates can be accessed from the -1 template with `<X-FC-OBJURL>?folders`. When the -1 template is rendered, the `<X-FC-OBJURL>` tag is replaced by the URL to the current item:

- -200 the subcontainer list header
- -201 the subcontainer list item (one for each folder)
- -202 the subcontainer list footer.

## Leaf item lists

These templates output container leaf listings with certain filters. This means that if you use these templates, only leaf objects are listed.

You can access these templates from the -1 template with `<X-FC-OBJURL>?folders`. When the -1 template is rendered, the `<X-FC-OBJURL>` tag is replaced by the URL to the current item:

- -210 the header for the leaf list
- -211 the leaf list item (one for each message)
- -212 the footer for the leaf list.

## Combined item lists

These templates output a complete listing of objects in a container with no filters at all. This means that if you use these templates, an integrated listing of both leaf objects and containers is generated.

These templates can be accessed from the -1 template with `<X-FC-OBJURL>?both`. When the -1 template is rendered, the `<X-FC-OBJURL>` tag is replaced by the URL to the current item:

- -205 the header for the combined item list
- -206 the combined item list entry (one for each object in the list)
- -207 the footer for the combined item list.

## Directory lists

These templates output Directory listings (from a Directory search) to the web, and are accessed from the `/Login/Directory` or `/Login/Directory/Lookup` paths in a URL. You must code these paths into the template for the URL to open it:

- -100 the Directory form (always output)
- -101 the Directory list header (output after the -100 when a Directory search has been initiated (the second URL))
- -102 the Directory list item (output for each element of a Directory listing)
- -103 the Directory list footer (output after the -101 and all the -102s in a Directory listing).

## History lists

The `<X-FC-OBJURL>` tag represents the URL to the current object, and `^H` means "open history for this object". You can access these templates from a message form with `<X-FC-OBJURL>^H`. When a message form is rendered, `<X-FC-OBJURL>` is replaced by the URL to the current history list. You must code these paths into the template for the URL to open it:

These templates output history listings to the web:

- -110 the history list header
- -111 the history list item
- -112 the history list footer.

## Search lists

These templates output search listings to the web.

You can access these templates from URL/search, to search everything at or below the URL. You must code this path into the template for the URL to open it:

- 8000            the search for:
- -300           the search results header
- -301           the search results item
- -302           the search results folder (output for every folder searched)
- -303           the search results folder.

## Who's Online lists

These templates output listings of users who are currently logged into the server. You can access these templates from /Login/Who's Online. You must code this path into the template for the URL to open it:

- -400           the Who's Online header
- -401           the Who's Online item
- -402           the Who's Online footer.

## Calendar lists

These templates output calendar listings to the web:

- -158           the calendar monthly view
- -159           the calendar weekly view
- -160           the calendar daily view (obsolete, replaced in most cases by the daytimer view template -167).
- -164           the calendar "today" view
- -167           the calendar daytimer view.

## Instant Messaging

These templates are used to output Instant Messaging and Instant Messaging transcripts

- 129            read-only chat transcript
- Edit.129      active (for example, in progress) Instant Messaging.

## Calendar events and tasks

These templates output event and task forms to the web:

- 161            the calendar event (read only)
- Edit.161      the calendar event (edit)
- 163            the calendar task (read only)
- Edit.163      the calendar task (edit).

## Messages

These templates output editable message forms to the web.

- 141            the standard message form (read only)
- Edit.141      the standard message form (edit).

[Top](#)