



Using Session Tags (S Tags)

Table of Contents

S Tags – Quick Reference	2
What Are S Tags?	9
S Tag Syntax	9
Adding S Tags Automatically in the WYSIWYG Editor	10
Coding S Tags in the Source Editor	10
Adding Comments to S Tags	11
Conditional S Tag Constructs	12
Structuring Conditionals	14
S Tag Library	20
Appendix A: S120 Object Properties	206
Donation2 (General Donations) Autoresponders	206
TeamRaiser Autoresponders	206
eCommerce Autoresponders	206
Tribute Autoresponders	207
Thank You Pages	207
List of S120 Objects	208
Appendix B: S Tags by Application	213
Appendix C: List of Application IDs to Use with S11	216

S Tags – Quick Reference

S Tag	Application	Purpose	CMS Compatible
S0	Platform	Site data	Yes
S1	Contacts	Displays user data for a logged-in user	Yes
S2	Navigation Bar	Insert a navigation bar	Yes
S3	Platform	User login/logout area	Yes
S4	Platform	Renders the current application ID	Yes
S5	Tell-A-Friend	Renders a tell-a-friend component	Yes
S6	Platform	Returns the base URL	No
S7	Platform	Renders the links Related to the currently-displayed content item	No
S8	Platform	Returns the current URL	Yes
S9	Platform	Displays the current date and time	Yes
S11	Platform	Returns the URL for the requested application	Yes
S15	Donations	Displays a fundraising campaign status indicator	Yes
S16	Reports	Custom Report Results	Yes
S18	Contacts	Displays a personalized synopsis of different donor data	No
S19	Contacts	Returns the constituent's name	Yes
S20	Platform	Renders current server Time	Yes
S22	PageBuilder	Returns a PageBuilder calendar component	Yes
S24	PageBuilder	Renders the title of a PageBuilder or StoryBuilder page	Yes

S25	PageBuilder	User accessibility check	Yes
S26	Platform	Returns a quick registration component	No
S27	PageBuilder	Renders action alert component	No
S28	PageBuilder	Renders a survey component	Yes
S29	Platform	Renders site information (from site_url)	Yes
S32	StoryBuilder	Renders news articles	No
S35	TeamRaiser	Renders TeamRaiser a campaign status thermometer	Yes
S36	TeamRaiser	Renders lists of TeamRaiser participants, teams, or gifts	No
S37	Redirector	Renders a redirect URL to track click-throughs	No
S38	Communication	Renders an unsubscribe URL	No
S39	TeamRaiser	Renders a list of TeamRaiser campaigns	Yes
S42	TeamRaiser	Renders TeamRaiser campaign information	Yes
S43	TeamRaiser	Renders TeamRaiser Team Information	No
S44	TeamRaiser	Renders TeamRaiser Quick Search	Yes
S45	Contacts	Renders group membership indication	Yes
S47	TeamRaiser	Renders TeamRaiser Custom Component	No
S48	TeamRaiser	Renders TeamRaiser Participant Information	Yes
S50	Platform	Renders HTTP request header data	Yes
S51	PageBuilder	Renders content of PageBuilder page	Yes
S52	StoryBuilder	Renders news search component	Yes

S53	Platform	Renders security category prefix	No
S54	Rewards	Renders reward point information	Yes
S55	Platform	Renders a random number	Yes
S56	Advocacy	Renders the find representatives component	Yes
S57	Donations	Renders a premium selection component	Yes
S58	Communication	Renders an email message	Yes
S59	Platform	Renders date strings as milliseconds	Yes
S61	Facebook	Facebook Pages	Yes
S63	Platform	Renders Lorem Ipsum sample text	Yes
S65	Directory	Renders a directory quick search component	No
S66	Platform	Embeds another template	No
S67	Contributed Content	Qualified contribution	Yes
S68	Contributed Content	Specific contribution	Yes
S69	Contributed Content	Time contribution	Yes
S72	Platform	Renders localization (L10N) information	Yes
S75	Advocacy	Vote information	Yes
S79	Contributed Content	Solicited contribution	Yes
S80	Platform	Renders the value of a session variable	No
S81	Platform	Renders a sign-in component	Yes
S83	Centers	Renders information about the current center	Yes

S87	Platform	Displays string literals	Yes
S88	Platform	Renders a dynamic ask component amount	Yes
S90	Advocacy	Renders advocacy alert information	Yes
S93	Advocacy	Renders the user's representatives	Yes
S94	Advocacy	Renders an alert list	Yes
S96	Advocacy	Renders the alerts taken list component	Yes
S97	Platform	Renders the RSS link component	Yes
S98	Donations	Time after/time before a given date	Yes
S99	Platform	Renders a URL shortcut	Yes
S100	Platform	Site data parameter as template	No
S101	Platform	Days until password expiration	No
S102	Platform	Renders a visual alert (warning) box	Yes
S103	Constituent	Constituent attribute selection list	Yes
S120	Platform	Renders object property	Yes
S121	Platform	Renders map property	Yes
S151	PageBuilder	Renders whether a PageBuilder page exists	Yes
S188	Platform	Renders a dynamic ask component with minimum and maximum values	Yes
S201	Platform	Generate help topic link	Yes
S202	Platform	GIGYA login component	Yes
S203	Social Media	Social sharing component	Yes
S311	NavBar	Displays a navigation bar	Yes
S330	TeamRaiser	TeamRaiser participant check-in status	Yes
S334	Platform	Renders HTTP query data	No

S335	TeamRaiser	Campaign type	No
S336	Widgets	Banner field	No
S337	Widgets	Progress field	Yes
S345	TeamRaiser	Top 10 list for TeamRaiser national companies	Yes
S346	TeamRaiser	Renders a national company gift summary	Yes
S347	TeamRaiser	Renders TeamRaiser national company information	No
S348	TeamRaiser	Renders TeamRaiser dynamic progress	Yes
S349	TeamRaiser	Renders the total dollars raised for a TeamRaiser event in a dynamic way	Yes
S350	TeamRaiser	Renders TeamRaiser event manager permission	Yes
S355	Open Extensions	Google maps	No
S356	Open Extensions	YouTube video	Yes
S364	Open Extensions	YouTube video bar	Yes
S366	Facebook Connect	Renders a Facebook connect login component	No
S400	TeamRaiser	Renders the value of a TeamRaiser Specific Data Parameter (SDP)	No
S402	TeamRaiser	TeamRaiser horizontal navigation bar	No
S403	TeamRaiser	TeamRaiser program type tally	No
S404	TeamRaiser	Renders a Donor Wall with the 5 most recent donors of a team or event	Yes
S405	TeamRaiser	Renders a milestone value	Yes

S406	Reports	Show report writer results	No
S407	TeamRaiser	Show amount raised by TeamRaiser donations	Yes
S408	TeamRaiser	Renders a form (with results) to search for TeamRaiser events	Yes
S409	TeamRaiser	Shows a Participant Center 2 in a Convio custom page	No
S413	TeamRaiser	Renders a TeamRaiser eCommerce link	Yes
S414	TeamRaiser	Lists all affiliated Events for a given lead event	Yes
S415	TeamRaiser	Lists the top teams among affiliated events for a given lead event	Yes
S421	Personal Events	Renders a <i>get together</i> search field	Yes
S422	Personal Events	Displays a personalized link in the email sent to contacts of the host	Yes
S500	MultiCenter	Displays a center-specific data parameter (CDP)	Yes
S530	Personal Fundraising	Champion fund list	Yes
S535	Personal Fundraising	Personal fundraising campaign or fund Status thermometer	Yes
S542	Personal Fundraising	Personal fundraising campaign component	No
S548	Personal Fundraising	Champion information component. Similar to S48	No
S549	Personal Fundraising	Is champion of any fund tag? (TRUE/FALSE)	Yes
S550	UI Controls	Select menu listing years (in Donation and Payment pages)	Yes
S551	UI Controls	Absolute URL Component	Yes

S565	Communication	Renders list of active and sent email messages	Yes
S1151	Advocacy	List of Letter to the Editor alerts	Yes
S1152	Advocacy	List of Letter to the Editor alerts taken by the current user	Yes
S1153	Advocacy	List of Letter to the Editor alerts sent by logged-in Constituent	Yes

What Are S Tags?

S tags are *session tags* that allow you to import published content into a custom page. For example, with S tags you can:

- **Render** an action alert, a survey, donation thermometer, or reusable page content. This content is also dynamic. For example, the donation thermometer automatically adjusts as the donation amount changes.
- **Personalize** content for the logged-in user with information such as first name, last name; rewards point balance, or donation history.
- **Conditionalize** content for the logged-in user with information such as “is a member/is not a member”, “is an advocate/is not an advocate”, or “is a donor/is not a donor”

An S tag is a proprietary Convio tag that only renders on Convio Online Marketing (COM) or Content Management System (CMS) pages (and not on external pages). As such, you can add an S tag in either the COM and CMS WYSIWYG or source editors, and place it anywhere template rendering is used. Content retrieved with an S tag, which may be anything from a single integer to an entire template, can be derived from an application other than the one in which it is embedded.

Note: Some S tags do not render anything; rather, they return a value of "True" or "False." These tags are used exclusively for conditional tests.

S Tag Syntax

Most page authors can add S tag functionality without knowing the syntax or even what an S tag is. This is because the most commonly-used S tags can be added through the WYSIWYG editor (in which case the S tag is coded automatically). However, for authors who prefer to work directly in source mode, or who want to customize their site beyond the WYSIWYG functionality, an S tag can be coded as a bracketed, text-based variable, or as an XML tag.

As a bracketed, text-based variable

The bracketed, text-based variable syntax for an S tag is:

`[[Sx:parameter.parameter]]`

Where x is the S tag number and the parameters are separated by colons. For example, by adding the following parameters, the S42 tag renders the top 25 TeamRaiser fundraisers (where 1285 is the TeamRaiser ID and top-participants is the number of participants to list):

`[[S42:1285:top-participants:25]]`

If there are no parameters, do not add a colon. For example:

`[[S4]]`

As an XML tag

For authors who prefer to code in XML, the S tag syntax is:

```
<convio:session name="X" param="parameter1:parameter2"/>
```

where X is the S tag number and the parameters (if any) are separated by colons.

Note: In a few instances, an S tag developer chose to use an asterisk (*), a comma (,), a hash (#), or an underscore(_) -- rather than a colon (:) -- as the delimiter for S tag arguments. The variation is noted in this document's data for such session tags.

Adding S Tags Automatically in the WYSIWYG Editor

The most commonly-used S tags can be added through the WYSIWYG editor and require no knowledge of S tag syntax. The process to add differs depending on whether you are in the COM or the CMS product.

To insert S tags using the COM WYSIWYG

1. In the COM WYSIWYG editor, click **Convio Components**.

The Insert Convio Component menu displays.

2. Select a component and click **Insert**. The component options vary depending on which component you select.

The component displays in the CMS WYSIWYG editor.

To insert S tags using the CMS WYSIWYG

1. In the CMS WYSIWYG editor, click **Insert Component**.

2. Select **CRM Content** from the Component Type drop-down list.

The COM WYSIWYG editor displays.

3. In the COM WYSIWYG editor, click **Convio Components**.

The Insert Convio Component menu displays.

1. Select a component and click **Insert**. The component options vary depending on which component you select.

2. Click **Finish** to close the COM WYSIWYG editor.

The component displays in the CMS WYSIWYG editor.

Coding S Tags in the Source Editor

Authors who prefer to work directly in source mode, or who want to customize their site beyond the WYSIWYG functionality, must code S tags manually. The process differs depending on whether you are in the COM or the CMS product.

To code S tags directly in the COM source code

- Open the Plain Text Editor and code the S tag.

To code S tags directly in the CMS source code

1. In the CMS WYSIWYG editor, click **Insert Component**.
2. Select **CRM Content** from the Component Type drop-down list.
The COM WYSIWYG editor displays.
3. In the COM WYSIWYG editor, click **Use Plain Text Editor**.
The plain text editor displays.
4. Enter the code for the S tag and click **Apply**.
The plain text editor closes and the content displays in the CMS WYSIWYG editor.

Adding Comments to S Tags

Comments are notes that you can add to the source code, but that are not rendered in the browser. In the square bracket format, use an equal sign (=) followed by the text of the comment.

For example, the S19 tag renders the URL of an application whose ID is 19. The “PageServer” comment reminds you what application ID 11 refers to, but the word PageServer does not display in the browser.

```
[[S11:19=PageServer]]
```

The S20 tag inserts the server time. In this example, the comment “GMT” reminds the administrator that this is Greenwich Mean Time, but “GMT” does not display in the browser.

```
[[S20=GMT]]
```

Note: Do not add spaces before or after the “=”. For example, `[[S20 = GMT]]` is not the same as `[[S20=GMT]]`.

Conditional S Tag Constructs

Convio conditionals take one dynamic data value and check it against another to validate information that controls how content renders. The basic syntax is "If A matches B, then C, else D." The values might, for example, resolve to something like:

"If the logged in user's name *[a]* is John Smith *[b]*, show a list of all Jamestown team members *[c]*. Otherwise, just show a list of all teams *[d]*."

Bracket Syntax: `[[?value_a::value_b::template_c::template_d]]`

The following lists additional syntax rules:

- The whole conditional is enclosed in `[[? and]]`.
- The double colon `::` separates the individual values.
- The match is an exact, case-sensitive character match.

Using a Conditional to Customize User Greeting

This is an example of using a conditional to customize the user's greeting.

`[[?[[S1:first_name]]::George::Welcome back, George!::Welcome, new member!]]`

In English, this resolves to:

If the first name of the current user is George, show "Welcome back, George!" Otherwise, show "Welcome, new member!"

Constraining Exact Matches

`[[?Roberta::Robert::Match::No Match]]` will return "Match" because "Robert" is contained within (is a substring of) "Roberta."

In a conditional, constrain a test to an exact match by using delimiters around values. For example:

`[[?xxRobertxx::xxRobertxx::Match::No Match]]`

returns "No Match" because "xxRobertxx" is not an exact match of "xxRobertxx."

A double x (xx) is used in code samples in this document. Any delimiter can be used. To check for an empty value, use xx by itself. For example, to check if a user is logged in:

`[[?xx::x[[S1:user_name]]x::Not Logged In::Logged In]]`

If a user is logged in as *user_name RoadRunner*, the value from the *S1:user_name* tag is *RoadRunner*, which does not match xx. This causes the conditional to render the *else* statement "Logged In."

If a user is not logged in, the value from the *S1:user_name* tag is empty and `x[[S1:user_name]]x` resolves as xx when the template processes the S tag. This

results in a match in the conditional statement, prompting the render of the *then* clause of the conditional *Not Logged In*.

Structuring Conditionals

Simple Tests

- If the test case matches, do something, else do nothing:
`[[?xx[[S1:first_name]]xx::xxBobxx::Hi, Bob!::]]`
(If the first_name is Bob, render "Hi, Bob!" otherwise render nothing.)
- If the test case matches, do one thing, else do another thing:
`[[?xx[[S1:first_name]]xx::xxBobxx::Hi, Bob!::Hello.]]`
(If the first_name is Bob, render "Hi, Bob!" otherwise render "Hello.")

Compound Tests

If first name is Bob, say "Hi Bob!" Otherwise, if the first name is Frank, say "Hi Frank!" and if it is not Frank, just say "Hello."

```
[[?xx[[S1:first_name]]xx::xxBobxx::Hi
Bob!::[[?xx[[S1:first_name]]xx::xxFrankxx::Hi Frank!::Hello]]]]
```

OR Conditionals

If the first name is Mary or Misty or Janice, say "There is at least one lady present." Otherwise, say "Neither Mary nor Misty nor Janice is present."

```
[[?[[?[[S1:first_name]]::Mary::T::]][?[[S1:first_name]]::Misty::T::]]
[[?[[S1:first_name]]::Janice::T::]]::T::Hello Ladies!::Hello]]
```

Double Binds

A test for two conditions at the same time:

```
[[?[[?xxconst1xx::xxredxx::TRUE::FALSE]][?xxconst2xx::xxbluexx::TRUE:
:FALSE]]
::TRUETRUE::then-clause::else-clause]]
```

In this conditional, *const1* is tested for the presence of *red*. The value returned if a match is made is TRUE; otherwise, FALSE is returned. The same test is then run on *blue* in *const2*. When the tests are complete, possible combinations returned are:

Combination	Initiates
TRUETRUE	then-clause
TRUEFALSE	else clause
FALSEFALSE	else clause

FALSETRUE else clause

For example, if the visitor's full name is *Curious George*, we can offer a banana.

```
[[?[[?xx[[S1:first_name]]xx::xxCuriousxx::TRUE::FALSE]]  
[[?xx[[S1:last_name]]xx::xxGeorgexx::TRUE::FALSE]]::TRUETRUE  
:: Have a banana!:: Welcome, visitor.]]
```

Writing Conditionals

The process for writing conditionals is:

1. Develop your scenario.
2. Translate it into variables such as X, Y, etc.
3. Understand the key terms *if*, *then*, *else*.
4. Write the conditional code.

Step 1: Develop your scenario

Your organization wants to develop different content for constituents based on their donor status. Possible options are *Undefined*, *Non-Donor*, *Indirect Donor*, *Joint Donor*, *Donor*, and *Unknown*. Three sets of content are to be developed:

1. *Non-Donor* content should display for constituents who have never donated.
2. *Donor* content should display for constituents who have donated at any time in the past. *Donor* will include *Indirect Donor*, *Joint Donor*, and *Donor*.
3. *Unknown* content should display for constituents whose donor status is unknown or undefined.

This construct are feasible thanks to the S1 tag, which displays True or False when testing the donor status of the logged-in user.

Steps 2 and 3: Translate the scenario into variables and understand the key terms

If the logged in user is a non-donor, then display non-donor content (such as updated reasons to donate).

If the logged in user is a donor, then display active donor content (such as the great good achieved thanks to the most recent donations).

Otherwise, if the user's donor status is unknown or undefined, display a survey on attitudes toward donating.

- x is the non-donor S1 tag, and p is the content uniquely displayed to non-donors:

If x=TRUE then p,

- y is the undefined donors:
if y=TRUE then TRUE
- z is the unknown donors:
if z=TRUE then TRUE
- q - the content displayed to them.
- r is the content displayed to all donors:
else r

If the first condition is true, nothing after x=y then p will be tested or rendered.

If the first condition is false, then y=TRUE and z=TRUE will be tested. If the test returns anything other than FALSEFALSE, then q will be displayed.

If y and z test to FALSEFALSE, then the user is a donor and r will be rendered.

Step 4: Write the conditional code

Translate the scenario into variables when you write the code.

```

If x=TRUE then p,
else
if y=TRUE then TRUE else FALSE
if z=TRUE then TRUE else FALSE
if FALSEFALSE then r
else q
[[?x::TRUE::p
::
[[?
    [[?y::TRUE::TRUE::FALSE]]
    [[?z::TRUE::TRUE::FALSE]]
::FALSEFALSE::
r
q
]]
]]

```

The S1 tag returns TRUE if the logged-in user holds the donor status for which you test, and FALSE if not. It would be very complicated to test for all the permutations of TRUE/FALSE for the three types of donor (there are nine possible combinations). Testing for both unknown and undefined donor status that return FALSE is quite simple.

```
[[?[[S1:donor_status]]:: Non-Donor::Show Non-Donor Content::[[?
```



```
[[?[[S1: donor_status]]::Unknown::TRUE::FALSE]]
[[?[[S1:donor_status]]::Undefined::TRUE::FALSE]]
::FALSEFALSE::Show Donor Content (both unknown and undefined
donor status are false, so the user must be a donor)::Show
unknown/undefined donor status content.]]]]
```

Convio Tags: XML Syntax Conditionals

As previously mentioned, Convio supports an alternative, XML-based approach to dynamic customization known as *Convio tags*.

More elaborate than bracketed S tags, the XML version may nonetheless be more immediately accessible to administrators familiar with XML and XSL.

The following code sample demonstrates an "If ... else if ... else if ... else" construct.

```
<convio:choose>
  <convio:when ID="whatsontv" title="Affiliate" test="eq">
    <convio:op>PR_</convio:op>
    <convio:op><convio:session title="AffiliateSessionVar" name="80"
    param="Affiliate"></convio:session></convio:op>
    <convio:then>What's On TV?</convio:then>
  </convio:when>

  <convio:when title="events" test="eq">
    <convio:op>evt_</convio:op>
    <convio:op><convio:session title="AffiliateSessionVar" name="80"
    param="Affiliate"></convio:session></convio:op>
    <convio:then>Events</convio:then>
  </convio:when>

  <convio:when title="events" test="eq">
    <convio:op>2</convio:op>
    <convio:op><convio:session title="CurrentApp"
    name="4"></convio:session></convio:op>
    <convio:then>Events</convio:then>
  </convio:when>

  <convio:otherwise title="Unknown">Default Affiliate</convio:otherwise>
```

</convio:choose>

If the session variable *Affiliate* is *PR_*, then the *What's On TV?* displays and the choose completes. If, however, the first condition is not successful (if *Affiliate* is **not** *PR_*), the session variable *Affiliate* is checked for the value of *evt_*. If *Affiliate* is *evt_*, then the word *Events* displays. If the current application has an ID of 2 (Calendar), then the word *Event* displays. Finally, if none of these matches, the *Default Affiliate* displays.

Components of Convio Tags

<convio:choose></convio:choose>

Contains all the conditions and default "otherwise" content.

<convio:when></convio:when>

Contains a condition and the content to render if that condition is successful. Success is selected using the `case=""` attribute:

case="true"

This condition will be successful if it evaluates as true. This is the default behavior if no `case=""` attribute is present.

case="false"

This condition will be successful if it evaluates as false.

The type of comparison is determined by the value of `test=""`:

test="empty"

This will be true if the value is empty. Requires only one <convio:op></convio:op> element.

test="nempty"

This will be true if the value is not empty. Requires only one <convio:op></convio:op> element.

test="eq"

This will be true if the values match. Requires two <convio:op></convio:op> elements.

test="neq"

This will be true if the values do not match. Requires two <convio:op></convio:op> elements.

test="contains"

This will be true if the second value can be found somewhere in the first value. Requires two <convio:op></convio:op> elements.

test="ncontains"

This will be true if the second value cannot be found somewhere in the first value. Requires two `<convio:op></convio:op>` elements.

`<convio:op></convio:op>`

Contains a value to be compared. This may be text, HTML, or other `<convio>` / `[[Xn]]` tags.

`<convio:then></convio:then>`

Contains text to be rendered. This may be text, HTML, or other `<convio>` / `[[Xn]]` tags.

`<convio:otherwise></convio:otherwise>`

Contains the default text to be rendered if none of the `<convio:when>` elements are successful. This may be text, HTML, or other `<convio>` / `[[Xn]]` tags.

S Tag Library

Each Session tag contains the following information:

- **Description** - A general description of purpose of the tag
- **Application** - The application to which the tag pertains
- **WYSIWYG Equivalent** – The most commonly-used S tags are accessible through the WYSIWYG dropdown menus. For example, `[[S1:first_name]]` is the same as selecting "First Name" from the WYSIWYG Personalization menu.
- **CMS Compatible** – Whether or not the S tag is compatible with CMS. All S tags are compatible with COM (also known as CRM), but not all are compatible with CMS.
- **Format** - Correct S tag syntax in both standard, bracketed variable and XML format
- **Parameters** - The S tag's parameters, if any
- **Sample Code** – A usage example in both bracketed variable and XML format

S0 Site Data (Deprecated, see [S100](#))

Use the S0 tag to render the value of a Site Data Parameter (SDP).

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**

[[S0:SDP]]

- **XML**

<convio:session name="0" param="SDP" />

Parameters

SDP

Sample Code

- **Standard Input**

Embed the defined site title:

```
<title>[[S0:SITE_NAME]]</title>
```

Display a site data content warning:

```
[[?xy::x[[S0:USERNEWS_CONTENT_WARNING]]y:::<p  
class="Hint">[[S0:USERNEWS_CONTENT_WARNING]]</p>]]
```

- **XML Input**

```
<convio:session name="0" param="USERNEWS_CONTENT_WARNING" />
```

S1 User Data

Use the S1 tag to display user data for a logged-in user. The value rendered will be an attribute or preference for the current user (constituent). Interests are stored as preferences, so this tag can also return the user's preference for a specific interest category. For example, to greet the logged in user by first name:

Hello, [[S1:first_name]]!

Application

Platform

WYSIWYG Equivalent

Unlike most S tags, S1 tags are added in the WYSIWYG editor by clicking the **Convio Personalization** button, rather than clicking the **Convio Component** button.

To add an S1 tag in COM:

- Click **Personalization** and select any item except Rewards.

To add an S1 tag in CMS:

1. In the CMS WYSIWYG editor, click **Insert Component**.
2. Select **CRM Content** from the Component Type drop-down list.
The COM WYSIWYG editor displays.
3. In the COM WYSIWYG editor, click **Convio Personalization**.
The **Insert Convio Component** menu displays.
4. Select any item except Rewards, and click **Insert**.
5. Click Finish to close the COM WYSIWYG editor.

The component displays in the CMS WYSIWYG editor.

CMS Compatible

Yes

Format

- **Standard**
[[S1:attribute identifier]] or [[S1:special identifier]]
- **XML**
<convio:session name="1" param="attribute identifier" />

Parameters

An *attribute identifier* is the value of the *ID* attribute of an AttrCfg element in the SiteAttrCfg XML file. For example, [[S1:cons_id]] will be substituted with the constituent ID of the current user/constituent.

Note: this syntax is preferred over the second, *special identifier*, syntax.

The special identifiers are as follows:

Description	Attribute Identifier
constituent identifier	cons_id
user name for constituent	user_name
cleartext password for Constituent Note: Renders a link to reset password if new user login flow is enabled (USER_SECURITY_ENHANCEMENTS_ENABLED=true)	user_password
constituent title/honorific	cons_title (e.g. "Ms.")
constituent first/given name	first_name
constituent middle name	middle_name
constituent last name or surname	last_name
generational qualifier for constituent (e.g. "Jr.")	cons_suffix
professional qualifier for constituent (e.g. "PhD")	cons_prof_suffix
constituent's primary email address	primary_email
first street of constituent's preferred postal address	street1
second street of constituent's preferred postal address	street2
third street of constituent's preferred postal address	street3
city of constituent's preferred postal address	city
state/province of constituent's preferred postal address	stateprov
zip/postal code of constituent's preferred postal address	zip
country of constituent's preferred postal address	country

constituent's preferred phone	phone number
-------------------------------	--------------

Sample Code

- **Standard Input**

```
[[?xx::x[[S1:user_name]]x::Who are you?::Howdy [[S1:first_name]]!]]
```

If no user_name value is present (meaning the user is not logged in), the "xx" will return "True" and the text "Who are you?" will display. If a value of "False" is returned, then the S1 tag has a value and the user is logged in.

```
[[?[[S1:USE_ACCESSIBLE_ALTERNATIVES]]::TRUE::<p>Required.</p>::]]
```

Displays a warning that content is required if the user has set preferences to use accessible alternatives. Otherwise, nothing is displayed.

- **XML Input**

```
<convio:session name="1" param="membership_status"></convio:session>
```

Using the S1 Tag to See if a User is Logged In

The S1 tag allows you to see if a user is logged in and display content accordingly.

1. Open a PageBuilder page.
2. Insert the following content:

```
[[?xx::x[[S1:user_name]]x::You are not logged in.:You are logged in.]]
```

This code tests whether a given user is logged in or not. If the [[S1:user_name]] returns a value, it will be the user name, flanked by x's. Obviously, something like "xgreenRunnerx" will not match "xx" (there will only be a match if the tag returns nothing).

3. Save the document.
4. Visit the page as a guest visitor. Confirm that the message, "You are not logged in," displays.
5. Log in and visit the page as a logged in user. Confirm that the "You are logged in" message displays.

S2 Insert a Navigation Bar

Use the S2 tag to render a named navigation bar.

Application

Navigation Bar

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**

```
[[S2:string1:string2]]
```

- **XML**

```
<convio:session name="2" param="string1:string2"/>
```

Parameters

- *string1* - The name of the navigation bar
- *string2* - The optional style of rendering. Defaults to match string1. The styles, Vertical or Horizontal, are available as part of the initial template file set.

Sample Code

Standard Input

```
[[S2:TRNav:Vertical]]
```

S3 Log In/Log Out Component

Use the S3 tag to render a log in or log out component. The S3 tag can take the parameters "User," "Admin" or the name of a custom log-in style. No parameter will call the Login/Logout_component.tpt, except in the IE WYSIWYG HTML editor which will replace your empty string with the word "null." You must instead specify "User," "Admin" or the name of a customized pair of templates.

Note: The Login Component includes the S202 tag if Open Authentication is enabled.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**

[[S3:parameter]]

- **XML**

<convio:session name="1" param="attribute identifier" />

Parameters

style - The name of the style to use. Technically, this identifies which template to use: Login_Component_style.tpt or Logout_Component_style.tpt. The baseline template set supports styles "User," "Admin" and the empty style.

Sample Code

- **Standard Input**

[[?xx::x[[S1:user_name]]x::[[S3:User]]::[[S3:User]]]]

- **XML Input**

<convio:session name="3" param="Admin"></convio:session>

S4 Current Application ID

Use the S4 tag to access the current application ID.

Application

Platform

WYSIWYG Equivalent

Conditionals > Current Application

CMS Compatible

Yes

Format

- **Standard:**

[[S4]]

- **XML:**

<convio:session name="4" />

Parameters

None

Sample code

- **Standard Input**

This will show the script for FileUploadServer (1146) and PageServer (19).

```
[[?x1146x19::x[[S4]]x::<script>alert('Only ascii format uploads supported!');</script>::]]
```

- **XML Input**

```
<convio:when title=Calendar  
test="contains"><convio:op>x2x21x25x</convio:op><convio:op>x  
<convio:sessionname="4"param=""></convio:session>x</convio:op>  
<convio:then></convio:then></convio:when><convio:otherwise title=Other>  
</convio:otherwise>
```

S5 TELL-A-FRIEND LINK

Use the S5 tag to offer users a text or image link to a form for emailing others about your site. The email form can optionally be displayed within the same browser window or in a pop-up window. This tag is also the only mechanism by which you can link to customized Tell-A-Friend content that you create specifically for PageBuilder embedding.

Application

Tell-A-Friend

WYSIWYG Equivalent

None

Note: The option under **Links** imports a generic TAF page, not the customized page.

CMS Compatible

Yes

Format

- **Standard**
[[S5:url_only]]
- **XML**
<convio:session name="5" param="url_only" />

Parameters

- *mfc_full_page* - The form opens in a new window when a user clicks the link. This is the value used if no parameter is specified.
- *mfc_popup* – The form displays in a pop-up window when a user clicks the link.
- *url_only* - Renders the URL to the Tell-A-Friend form (without the HTML that displays an image or makes the URL an active link).

Sample Code

- **Standard Input**
[[S5:url_only]]
- **XML Input**
<convio:session name="5" param="url_only"></convio:session>

S6 Base URL

Use the S6 tag to display the base URL of your site. This is used in the page wrapper to provide a base for relative links.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**

[[S6]]

- **XML**

<convio:session name="6" />

Parameters

None

Sample Code

In this example, if the base URL contains "Preview" in it, the onsubmit event handler and its Javascript call will NOT be rendered inside the <form> tag.

- **Standard Input**

```
<form[[[?[[[S6]]::Preview:::onsubmit="calculate(this);return  
shouldISubmit(this.tix_quantity.value);"]]]>
```

- **XML Input**

```
<convio:session name="6"></convio:session>
```

S7 Related Links

Use the S7 tag to display links related to the currently displayed content.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**

[[S7]]

- **XML**

<convio:session name="7" />

Parameters

None

Sample Code

- **Standard View**

[[S7]]

- **XML Input**

<convio:session name="7"></convio:session>

S8 Current URL

Use the S8 tag to access the current URL. Useful for conditionalizing wrapper elements based on what section of the site the user is on.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**

[[S8]]

- **XML**

<convio:session name="8"></convio:session>

Parameters

None

Sample Code

- **Standard Input**

In this example, the URL is extended with the appropriate argument to load the printer friendly version of the current page:

```
<a href="[[S8]][[?[S8]::?:&::?]]printer_friendly=1">Printer Friendly</a>
```

- **XML Input:**

```
<convio:session name="8"></convio:session>
```

S9 Current Date and Time

Use the S9 tag to access the current date and time.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**

[[S9:parameter]]

- **XML**

<convio:session name="9" param="parameter"></convio:session>

Parameters

style

Appropriate style parameters are template names:

CalDateTime	9/26/2001 4:37 PM
CalendarList	Sep 26, 2001
cons	September 26, 2001
default	Sep 26, 2001 4:37:25 PM
DMY	9/26/01
MY	Sep_01
news	Wednesday September 26, 2001
Survey	9/26/2001
timestamp	1190842645805
YMD	9/26/01

pattern

The special style pattern does not use a template, but rather accepts an additional parameter with a formatting pattern. This style should only be used for client customizations, as it lacks the consistency and semantics of the template-based styles.

A template customization can be done to customize this tag further. A DateOutput_foo.tpt template has access to these application arguments.

App Arg Number	Sample Value	Description
0	Jun 26, 2002 10:04:27 PM	full date
1	2002	four digit year
2	02	two digit year
3	June	month name
4	Jun	short month name
5	6	numeric month
6	26	day of month ?
7	26	day of month
8	177	day of year (Julian date)
9	Wednesday	day name
10	Wed	short day name
11	4	day of week (Sunday = 1)
12	4	day of week in month (on Sunday: first Sunday = 1, fifth Sunday = 5)
13	10	hour (12 hour clock)
14	22	hour (24 hour clock)
15	PM	AM or PM
16	04	minutes

17	27	seconds
18	448	milliseconds
19	CST	time zone
20	-600	GMT Offset
21	1025147067448	milliseconds since epoch (1/1/1970 00:00:00.000 GMT)

Sample Code

- **Standard Input**

The current date and time is:

```
[[S9:CalDateTime]]
```

Display month name and 4-digit year separated by a single space (for example, "July 2008"):

```
[[S9:pattern:MMMM yyyy]]
```

Display dd/mm/yyyy (for example, "01/07/2008"):

```
[[S9:pattern:dd/MM/yyyy]]
```

Display 4-digit year:

```
[[S9:pattern:yyyy]]
```

- **XML**

The current date and time is:

```
<convio:session name="9" param="CalDateTime"></convio:session>
```

S11 URL of Specified Application ID

Use the S11 tag to access the URL of an application from its numeric ID.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**

[[S11:applID]]

[[S11:applID:contextType]]

- **XML**

<convio:session name="11" param="applID"/>

Parameters

- *applID* - Numeric ID of any application for which URL is desired. Appendix C is a list of application IDs.
- *contextType* – The context type can be one of the following (email is the default)
 - **Page:** Returns the URL as a string that is suitable for inclusion in a web page. This means that it is HTML encoded (for example, ampersands are included as &) and may include an application context node and may include the session ID parameter JServSessionId. It also may be a relative URL rather than an absolute one.
 - **Unformatted:** Returns the URL as a string that does not have any extra formatting or session information. This means that it is not HTML encoded and does not include the session ID parameter JServSessionId.
 - **Redirect:** Returns the URL as a string that is suitable for an HTTP redirect. It may be a relative URL and may include an application context node ID and may include the session parameter JServSessionId.
 - **Email:** Returns the URL as a string that is suitable for inclusion in an email. This means that it does not include the session ID parameter JServSessionId. It is an absolute rather than relative URL. It may be HTML encoded depending on the rendering mode of the templateContext.

- **Editor:** Returns the URL as a string that is suitable for inclusion in a content editor such as the Convio WYSIWYG HTML editor.

Sample Code

- **Standard Input**

```
<a href="[[S11:81=ConsProfileUser]]">Edit your Constituent
Profile</a>

<a href="[[S11:12]]" />Click to search for your representative.</a>

[[?x[[S11:12]x::null::Congress is not in session::<a href="[[S11:12]]"
/>Click to search for your representative.</a>]]
```

You can add additional parameters to the results of S11. If the system already has parameters added then there will be a “?” in the URL returned. One can test for that before adding further parameters.

```
<a href="[[S11:19]][[?[[S11:19]]::?::&::?]]pagename=foo">go to a
certain page</a>
```

In the above example, The “[[?[[S11:19]]::?::&::?]]” will render a “&” if the PageServer URL already has a “?” (as it will if there is a jservsession parameter). Otherwise, the conditional will render a “?”.

A link to change password:

```
<a
href="[[S11:81=ConsProfileUser]][[?[[S11:81]]::?::&::?]]op.dispCon
sPasswd=">Change your password</a>
```

Note: Parameters can be added to the results of S11. If the system already has parameters added then there will be a “?” in the URL returned. Test for that “?” before adding further parameters.

[[S11:19]] renders "http://bvt301.convio.com/site/PageServer" on the "bvt301" test site.

[[S11:81:page]] renders "https://bvt300-secure.convio.com/bvt301/site/ConsProfileUser?JServSessionIda005=7yby9ytro1adm8009" with session ID

- **XML:**

```
<convio:session name="11" param="12" />
```

Commonly used with S11 to render a URL for a link that includes any session data that the application might want.

Note: For more information, see Appendix C: List of Application IDs.

S15 Fundraising Campaign Status Indicator

Use the S15 tag to render a specific fundraising campaign's status indicator components. Those components include:

- Progress Thermometer
- Campaign Title
- Campaign Goal
- Current Amount Raised
- Percentage of Goal Raised

You can also extract the distinct value from each of the above components separately.

WYSIWYG Equivalent

None

Application

Donations

CMS Compatible

Yes

Format

- **Standard**
[[S15:campaign_ID]]
- **XML**
<convio:session name="15" param="campaign_ID" />

Parameters

Campaign ID	Value
S15:1001	[Thermometer]
S15:1001:thermometer	[Thermometer]
S15:1001:title	(for example, Bugs)
S15:1001:goal	(for example, 5000000)
S15:1001:amountRaised	(for example, 436600)

S15:1001:percentRaised	(for example, 8)
S15:1001:goal_formatted	(for example, \$50,000.00)
S15:1001:amountRaised_formatted (for example, \$4,366.00)	
S15:1001:percentRaised_formatted	(for example, 8%)

Sample Code

- **Standard Input**

[[S15:2344]]

- **XML**

<convio:session name="15" param="2344" />

S16 Custom Report Results

Use the S16 tag to render the results of a custom report.

Application

Reports

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S16:report_id:false:a3:a4]]
- **XML**
<convio:session name="16" param="id,false,a3,a4" />

Parameters

report_id

The numerical report ID of the custom report to call in

False

This is required.

A3

This is passed to the custom report's SQL file as [[A3]].

A4

This is passed to the custom report's SQL file as [[A5]].

Sample Code

- **Standard Input**
[[S16:1000:false:Test:1]]
This calls custom report 1000 and sends in *Test* as [[A3]] and *1* as [[A4]] (for S16 tab A0=database prefix, A1 = site id, A2=timestamp).
These values would be used in the custom report SQL file.

The purpose and quantity of each argument varies from among custom reports. See the report's documentation for details.

- **XML**

```
<convio:session name="16" param="1000:false:Test:1" />
```


S18 Personalized Donor Data Synopsis

Use the S18 tag to render a personalized synopsis of donor data.

Application

Contacts

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**
[[S18:type]]
- **XML**
<convio:session name="18" param="type" />

Parameters

- *pledge_reminder* - displays a table of the users pending pledge payments
- *ytd* - displays user donations for this year
- *total* - displays the user's total donations
- *last* - displays the value of the user's last donation
- *largest* - displays the value of the user's largest donation

Sample Code

- **Standard Input**
[[S18:largest]]
- **XML**
<convio:session name="18" param="largest" />

S19 User's Name

Use the S19 tag to render the current user's name. The parameter is the constituent ID.

Application

Contacts

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**

[[S19:cons_ID]]

- **XML**

<convio:session name="19" param="cons_ID" />

Parameter

cons_id – The numeric constituent ID.

Sample Code

```
[[?xxnullx::x[[S19:1001141]]x::Welcome Friend::Welcome [[S19:1001141]]]]
```

S20 Current Server Time

Use the S20 tag to render the current time on the hosting server. The value will be formatted in Greenwich Mean Time:

Fri Sep 4 14:30:06 CDT 2009

Application

Platform

WYSIWYG Equivalent

None

Format

- **Standard**

[[S20]]

- **XML**

<convio:session name="20" />

CMS Compatible

Yes

Parameters

None

Sample Code

- **Standard Input**

The current date and time is [[S20]].

- **XML**

<convio:session name="20"/>

S22 Calendar Component

Use the S22 tag to pre-render a PageBuilder page with a calendar component.

Application

PageBuilder

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

Note the use of the underscore character as a separator, rather than the typical colon.

- **Standard**

```
[[S22: 1_0_2_-1_test_0_true_0]]
```

- **XML**

```
<convio:session name="22" param="1_0_2_-  
1_test_0_true_0"></convio:session>
```

Parameters

style

- 1 = Event Name List
- 2 = Small Grid
- 3 = Event Name and Description List
- 4 = Large Grid

period

- 0 = Current
- 1 = Next

periodlength

- 0 = Month
- 1 = Week
- 2 = Day

interest

Event Interest Category to display:

-1 = Show all

0 = Show favorites

Interest Category ID = Only events in that Interest Category

title

Any string

security

Security Category to display

0 = Show all

Security Category ID = Only events in that Security Category

showpast

Show past events

true = Show events in the past

false = Only show events in the future

maxdays

Maximum numbers of days to show

0 = No maximum

other number = Maximum number of days to display

Calendar_Style

1 = Event Name List

2 = Small Grid

3 = Event Name and Description List

4 = Large Grid

Time_Period

0 = Current

1 = Next

Time_Period_Length

0 = Month

1 = Week

2 = Day

Event Interest Category to display

-1 = Show all

0 = Show favorites

Interest Category ID

Only events in that Interest Category
Title To Display Above Calendar
Any string
security category to display
Only events in that Security Category
Show past events
true = Show events in the past
false = Only show events in the future
Max Numbers of Days to Show
0 = No maximum

Sample Code

- **Standard Input**

```
[[S22:1_0_2_-1_test_0_true_0"]]
```

- **XML**

```
<convio:session name="22" param="1_0_2_-  
1_test_0_true_0"></convio:session>
```

S24 Pagebuilder or Storybuilder Page Title

Use the S24 tag to create a breadcrumb effect within a PageBuilder or StoryBuilder page.

Application

PageBuilder

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S24]]
- **XML**
<convio:session name="24" />

Parameters

None

Sample Code

- **Standard Input**
[[S24]]
- **XML**
<convio:session name="24" />

S25 User Accessibility Check

Use the S25 tag to determine a user's accessibility preference. This feature is a shorthand version of

```
[[?x[[S1:USE_ACCESSIBLE_ALTERNATIVES]]x::xTRUEx::true_render_somethi  
ng::false_or_not_set_value_render_something_else_or_differently]]
```

The main use of this tag is to present content in a different configuration to a user who prefers accessible alternatives.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**

```
[[S25:true_rendering:false_rendering]]
```

- **XML**

```
<convio:session name="25" param=" true_rendering:false_rendering "/>
```

Parameters

- *accessiblity_flag_true_value* - value to be rendered if USE_ACCESSIBLE_ALTERNATIVES is TRUE for the current user.
- *accessiblity_flag_false_value* - value to be rendered if USE_ACCESSIBLE_ALTERNATIVES is **not** TRUE for the current user.

Sample Code

Standard Input

```
[[S25:<p>Required.</p>:<p>Not Required.</p>]]
```


S26 Quick Registration Component

Use the S26 tag to render a quick registration component.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**

```
[[S26:String1:Int:Bool1:Bool2:String2:String3:String4]]
```

- **XML**

```
<convio:session name="26"  
param="String1:Int:Bool1:Bool2:String2:String3:String4]"/>
```

Parameters

String1 – Tag syntax

V3

Int (for example, to display style)

0 - Popup form with name, email, and zip / postal code

1 - Embedded form with name and email

2 - Embedded form with email only

3 - Embedded form with email and zip / postal code

4 - Popup form with name and email

5 - Popup form with email only

6 - Popup form with email and zip / postal code

7 - Popup form with name, address, and email

8 - Popup form with name, state, and email

Bool1

Display Email Interests in Popup Form

TRUE = do

FALSE = do not

Bool2

Render HTML

TRUE = Render JavaScript and escaped HTML for cut-n-paste into a static page.

FALSE = Do not render

String2

Destination After Form

PROFILE = Take user to ConsProfileUser

INTERESTS

Take user to ConsInterestsUser

None

Take user directly to URL specified

String3

Button Text

The text on both the embedded and popup forms.

String4

Next URL

A valid URL that has been URL-encoded. The user will be taken to this page after completion of the form and Cons page.

Sample Code

- **Standard Input**

```
[[S26:V3:1:FALSE:TRUE:None:CONTINUE:http://www.convio.com]]
```

- **XML**

```
<convio:session name="26"  
param="V3:1:FALSE:TRUE:None:CONTINUE:http://www.convio.com"></con  
vio:session>
```

S27 Action Alert Component

Use the S27 tag to render an action alert component.

Application

PageBuilder

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**

[[S27:alert_ID*version_ID]]

- **XML**

<convio:session name="27" param="alert_ID*version_ID" />

Parameters

- *alert_ID* - The ID of the action alert component to render
- *version_ID* - The ID of the page version with which the component is associated

Sample Code

- **Standard Input**

[[S27:103*9]]

- **XML**

<convio:session name="27" param="103*9"></convio:session>

S28 Survey Component

Use the S28 tag to render a survey component.

Application

Survey

WYSIWYG Equivalent

Components > Survey

CMS Compatible

Yes

Format

- **Standard**

```
[[S28:EMBED:TITLE:GREETING:FIRST_QUESTION:RANDOM_QUESTION:REPORTING:Survey_ID:{Question_ID}]]
```

- **XML**

```
<convio:session name="28" param="EMBED:TITLE:GREETING:FIRST_QUESTION:RANDOM_QUESTION:REPORTING:Survey_ID:{Question_ID}" />
```

Parameters

- *embed* - display entire survey
- *title* - display survey title
- *greeting* - display survey greeting
- *first_question* - display first question from survey
- *random_question* - display a randomly selected question from the survey
- *reporting* - display the current results for the question indicated in question_ID (note that this is the question ID number, not the question number)

Sample Code

- **Standard Input**

```
[[S28:EMBED:1006]]
```

- **XML**

```
<convio:session name="28" param="EMBED:1006"></convio:session>
```

S29 Site Information

Use the S29 tag to render different types of site information. This includes the user ID, the domain, session ID, and the remote address.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**

[[S29: parameter]]

- **XML**

<convio:session name="29" param="parameter" />

Parameters

- *site_ID* - Renders site ID
- *session_ID* - Renders session ID
- *user_ID* - Renders user ID
- *domain* - Renders domain name of the site
- *path* - Renders current path
- *secure_domain* - Renders domain name of secure server for site
- *secure_path* - Renders secure path
- *admin_domain* - Renders domain name of administrative server for site
- *admin_path* - Renders current administrative path
- *request_domain* - Renders the host name used for the current request
- *original_domain* - Renders the original host name used for the current request before redirection
- *remote_addr* - Returns the IP address of the client machine in dotted quad format (that is, for example, 203.163.198.221)

Sample Code

- **Standard Input**

```
<A href="http://[[S29:DOMAIN]]/site/ConsProfileUser">Printer Friendly</A>
```

and

```

```

- **XML**

```
<convio:session name="29" param=" REMOTE_ADDR " />
```

S32 Storybuilder Articles

Use the S32 tag to render a grid of news articles.

Application

StoryBuilder

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**

```
[[S32:showStyle_securityCatId_maxRows_layoutStyle_nbrOfColumns_
interestCatID_showActive_showExpired]]
```

- **XML**

```
<convio:session name="32" param="
showStyle_securityCatId_maxRows_layoutStyle_nbrOfColumns_
interestCatID_showActive_showExpired " />
```

Parameters

- *ShowStyle:*

- 1 SHOW_HEADLINE
- 3 SHOW_ENTIRE_STORY
- 5 SHOW_HEADLINE_TEASER
- 4 SHOW_HEADLINE_TEASER_PHOTO
- 6 SHOW_HEADLINE_NOBULLET
- 7 SHOW_HEADLINE_DATE
- 8 SHOW_HEADLINE_DATE_BYLINE
- 9 SHOW_HEADLINE_DATE_TEASER
- 10 SHOW_HEADLINE_DATE_BYLINE_TEASER
- 11 SHOW_HEADLINE_DATE_TEASER_PHOTO
- 12 SHOW_HEADLINE_DATE_BYLINE_TEASER_PHOTO

- *securityCatId* – The security category filtering ID (0 for any / all categories)
- *maxRows* - The maximum number of rows (articles)

- *layoutStyle* - tall | wide
- *nbrOfColumns* - The number of article columns to render
- *interestCatID* - The interest category to filter by (0 for no interest specified)
- *showActive* - *Whether or not to include active articles.*
 true - includes active articles
 false - excludes active articles
- *showExpired*
 true - includes cataloged articles
 false - excludes cataloged articles
- *locale* - the locale filter to show only articles in that locale. For this specific application, use es-US, fr-CA, en-CA, or en-US. Use all to include all locales.

Sample Code

- **Standard Input**

```
[[S32: 1_0_100_tall_2_0_true_true]]
```

- **XML**

```
<convio:session name="32"  
param="1_0_100_tall_2_0_true_true"></convio:session>
```


S35 TeamRaiser Thermometer

Use the S35 tag to render a TeamRaiser Campaign Status Thermometer.

Note: If the value of *frID* is a Personal Fundraising campaign, or if the *teamID* represents a Personal Fundraising fund, use the S535 tag.

Application

TeamRaiser

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**

[[S35: type,frID,teamID,consID]]

- **XML**

<convio:session name="35" param=" type,frID,teamID,consID" />

Parameters

- *type*:
 - campaign_therm
 - team_therm
 - participant_therm
- *frID* - the ID of the TeamRaiser
- *teamID* - the ID of a team in the TeamRaiser
- *consID* - the ID of a TeamRaiser participant

Sample Code

- **Standard Input**

[[S35: team_therm,1001,1122,1040]]

- **XML**

<convio:session name="35" param=" team_therm,1001,1122,1040" />

S36 Lists of TeamRaiser Participants, Teams, or Gifts

Use the S36 tag to render a list of TeamRaiser Participants, Teams, or Gifts. Typically, this is a ranking, as of the top 5 teams. Can be used to insert the status indicator in the WYSIWYG on a team page.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**

```
[[S36:type,fr_ID,team_ID,cons_ID,display_mode,display_amt,display_size,title]]
```

- **XML**

```
<convio:session name="36"  
param="type,fr_ID,team_ID,cons_ID,display_mode,display_amt,display_si  
ze,title" />
```

Parameters

- *type*
 - top_teams_list
 - top_participants_list
 - top_gifts_list
 - top_5_teams_by_div_list
 - top_10_teams_by_div_list
 - top_5_participants_by_type_list
 - top_10_participants_by_type_list
 - top_team_gifts_list
 - top_events_by_affiliate_list
 - team_donor_list
- *fr_ID* - the ID of the TeamRaiser
- *team_ID* - the ID of a team in the TeamRaiser

- *cons_ID* - the ID of a TeamRaiser participant
- *display_mode* , which is one of:
 - scrolling
 - list
 - textarea
 - editmode
- *display_amt*, which is one of:
 - sum
 - count
- *display_size* - a number specifying the maximum number of items to display in the list
- *title* - the title to display at the top of the list.

Sample Code

Example of adding an honor roll to a PageBuilder page where the Teamraiser's ID is 1040:

- **Standard Input**

```
[[S36:top_teams_list,1040,0,0,SCROLLING,SUM,100,TOP TEAMS]]
```

Because the team_ID and cons_ID are unknown, "0" makes a call for all teams and all cons within the TeamRaiser.

- **XML**

```
<convio:session name="36" param="
top_teams_list,1040,0,0,SCROLLING,SUM,100,TOP TEAMS " />
```

S37 Track Click-Throughs Not for Clients

Use the S37 tag to render redirect URL to track click-throughs.

Application

Redirector

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**

[[S37]]

- **XML**

<convio:session name="37" />

Parameters

None

Sample Code

- **Standard Input**

[[S37]]

- **XML**

<convio:session name="37" />

S38 Unsubscribe URL

Use the S38 tag to render an unsubscribe URL when the user clicks to unsubscribe.

Application

Communication

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**
[[S38]]
- **XML**
<convio:session name="38" />

Parameters

None

Sample Code

- **Standard Input**
[[S38]]
- **XML**
<convio:session name="38" />

S39 List of TeamRaiser Campaigns

Use the S39 tag to render a list of TeamRaiser campaigns. If an Interest is indicated in the tag parameters, then the data will be displayed in a table with the date of the event as configured on the TeamRaiser greeting page. If an Interest is not specified, a list of data will be displayed.

The S39 tag outputs *public event type* data if the related site option is enabled.

If an Interest is selected to filter the TeamRaisers, the data will display in a table with the date of the event as configured on the TeamRaiser greeting page. If no interest is selected, it will be a list of the TeamRaiser names.

Will only list TeamRaiser Events with status of:

- Accepting Gift
- Accepting Registrations
- Gifts Accepting Registrations Only.

If we are displaying a list of events managed by the logged-in constituent, we will include Unpublished and Closed.

Application

PageBuilder

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**

```
[[S39:interest*colour*size*useBullets*maxListSize*displayType*security*title:showOnlyRegistered:type:fiscalYear]]
```

- **XML**

```
<convio:session name="39"
param="interest*colour*size*useBullets*maxListSize*displayType*security*title:showOnlyRegistered:type:fiscalYear" />
```

Parameters

- *interest* - the ID of the interest category to filter campaigns (use 0 to disregard interests)
- *colour* - hexadecimal (#000000) or named color (black)

- *size* - the size of the scalable font to use
- *useBullets* - true or false
- *maxListSize* - the maximum number of campaigns
- *displayType* - 0 to show all campaigns, and 1 to show campaigns filtered by interest
- *security* - 0 to show all campaigns, or the security cat ID filter
- *title* - the text for the title of the list
- *showOnlyRegistered* - 'true' or 'false'
- *type* - default is blank, use a string to filter based on this TeamRaiser type
- *fiscalYear* - default is blank, if a year (2006) filters list on TeamRaiser fiscal year
- *showOnlyEventsManaged* - default is false, if true, this will cause the list to render only events that are managed by the logged in constituent

Sample Code

- **Standard Input**

```
[[S39:0*red*14pt*true*50*0*0*TeamRaisers]]
```

- **XML**

```
<convio:session name="39"
param="0*red*14pt*true*50*0*0*TeamRaisers"></convio:session>
```

S42 TeamRaiser Campaign Information

Use the S42 tag to render summary information for a TeamRaiser.

Note: If the value of *frID* is a Personal Fundraising campaign, use the S542 tag.

Application

TeamRaiser

WYSIWYG Equivalent

Components > TeamRaiser Components

CMS Compatible

Yes

Format

- **Standard**

[[S42:fr_ID:type:param]]

- **XML**

<convio:session name="42" param=" fr_ID:type:param " />

Parameters

fr_ID - the ID of the TeamRaiser

type, one of the following:

affiliate-type - The type of event affiliation for this TeamRaiser.

city - The city of the event.

classification - Primary Event Type for the TeamRaiser.

classification2 - Secondary Event Type for the TeamRaiser.

days-to-event - Days until the event.

dollars - The total amount of donations that have been made to the TeamRaiser on behalf of a participant.

ecommerce-content - eCommerce page content that the event manager can specify.

emc-url - The URL for event management center for this event.

event-date - Event Date for the TeamRaiser.

form-ID - The ID of the Donation form associated with the TeamRaiser.

goal - The fundraising goal for the TeamRaiser.

lead-fr-id - The ID of the TeamRaiser.

local-officemailing-address - The local event mailing information configured under the location information.

num-participants - The number of participants registered for the TeamRaiser.

num-teams - The number of teams created for the TeamRaiser.

num-active-badges - The number of badges that are configured to be active for the event (new in Aspen).

num-donors - The number of people that have donated to the TeamRaiser on behalf of a participant.

num-donations - The number of donations that have been made to the TeamRaiser on behalf of a participant.

param1 - a type-specific parameter used thusly: for the top-teams, top-companies, top-companies-rollup and top-participants types, the number of entries to include in the list.

participation-types-available - If registration limits are enabled, returns TRUE or FALSE to indicate whether any participation types are available.

prev-fr-id - Previous Teamraiser ID, if available; otherwise null.

product-name - Name of the associated ecommerce product.

public-event-type - The name of the [public event type associated](#) with the TeamRaiser campaign.

registration-limit-type - The type of registration limit imposed on the event. Returns None, EVENT, or PARTICIPATION_TYPE.

state - The state of the event.

status - Numeric value for status of campaign (0 = unpublished, 1 = accepting registration, 2 = accepting reg & gifts, 3 = deny reg, accept gifts, 4 = closed, 8 = archived, 9 = deleted) .

store-name - Name of the associated ecommerce store.

street - The street address of the event.

tax-ID - The tax ID for the TeamRaiser.

title - Public Name for the TeamRaiser.

to-reach-goal - Amount to reach the fundraising goal.

top-companies - A list of the top performing companies in the TeamRaiser, based on the total amount of donations made on behalf of company teams. This type treats child companies (i.e. departments & facilities) as independent entities and does not roll child company data up to the root ancestor company.

top-companies-rollup - A list of the top performing companies in the TeamRaiser, based on the total amount of donations made on behalf of

company teams. This type treats donations made to child companies (i.e. departments & facilities) as belonging to the parent top-level company. This type is new as of Aspen and is only supported when precalc is enabled.

top-teams - A list of the top performing teams in the TeamRaiser, based on the total amount of donations made on behalf of team members.

top-participants - A list of the top performing participants in the TeamRaiser, based on the total amount of donations made on their behalf.

type - Returns campaign type, identical to [S335](#).

zip - The zip code of the event.

Sample Code

- **Standard Input**

Top 10 Teams

[[S42:1285:top-teams:10]]

Top 25 Individual Fundraisers

[[S42:1285:top-participants:25]]

- **XML**

<convio:session name="42" param="1285:top-teams:10"></convio:session>

S43 TeamRaiser Team Information

Use the S43 tag to render Team Information.

Application

TeamRaiser

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**

[[S43:fr_ID:type:param]]

- **XML**

<convio:session name="43" param=" fr_ID:type:param" />

Parameters

fr_ID

the ID of the TeamRaiser

type, contains one of the following:

company-id - The id of team's company; null if no company

company-name - The name of the team's company; null if no company

dollars - The total amount of donations made to the TeamRaiser on behalf of members of this team

goal-members - The team captain's goal for the number of participants to join this team

goal-dollars - The team captain's goal for the total amount of donations to be made on behalf of members of this team

name - The name of the team

num-members - The number of participants who actually joined this team

num-donors - The number of people who have donated to the TeamRaiser on behalf of a participant

param - a type-specific parameter, such as, for the *top_teams*, the number of teams to include in the list

rank - This team's rank, or position, within all teams in the same TeamRaiser, based on the total amount of donations made on behalf of team members

rank-div - This team's rank, or position, within the teams in the same division for the same TeamRaiser, based on the total amount of donations made on behalf of team members

top-teams - A list of the top performing teams in the TeamRaiser that are also in the same division as this team, based on the total amount of donations made on behalf of team members

team - An optional hard-coded team id, except in the case of *top_teams* where it is the number of teams to include in the list. If it is not given the team id will default to the current user's team. For example:

- S43:1:name gives the name of the current user's team if it exists in TeamRaiser event 1.
- S43:1:name:2 gives the name of team 2 if it exists in TeamRaiser event 1. Otherwise it tries the team of the current user as if a team number were not given.

Sample Code

- **Standard Input**

```
[[S43:1000:top_teams:5]]
```

- **XML**

```
<convio:session name="43" param="1000: top_teams:5"></convio:session>
```

S44 TeamRaiser Quick Search

Use the S44 tag to render a TeamRaiser Quick Search field. Participants are located if the search criteria appear in their team name or at the start of their first or last names.

Application

TeamRaiser

WYSIWYG Equivalent

Components > TeamRaiser Search > Quick Search

CMS Compatible

Yes

Format

- **Standard**
[[S44:fr_ID]]
- **XML**
<convio:session name="44" param="fr_ID" />

Parameters

fr_ID - The ID of the TeamRaiser

Sample Code

- **Standard Input**
[[S44:1000]]
- **XML**
<convio:session name="44" param="1000" />

S45 Group Membership Indicator

Use the S45 tag to confirm group membership. The tag returns "True" if the current, logged-in user is a member of the specified group. Otherwise it returns "False."

Application

Contacts

WYSIWYG Equivalent

Unlike most S tags, S45 tags are added in the WYSISYG editor by clicking the **Convio Conditional** button, rather than clicking the **Convio Component** button.

To add an S45 tag in COM:

- Click **Conditionals** and select **Groups > Is in group/Is not in Group**.

To add an S45 tag in CMS:

1. In the CMS WYSIWYG editor, click **Insert Component**.
2. Select **CRM Content** from the Component Type drop-down list.
The COM WYSIWYG editor displays.
3. In the COM WYSIWYG editor, click **Convio Conditional**.
The **Insert Convio Conditional** menu displays.
4. Select **Groups > Is in group/Is not in Group** and click **Insert**.
5. Click Finish to close the COM WYSIWYG editor.

CMS Compatible

Yes

Format

- **Standard**
[[S45:group_ID]]
- **XML**
<convio:session name="45" param="group_ID"/>

Parameters

group_ID - The ID of the group to check

Sample Code

- **Standard Input**

[[?[[S45:1001]]::TRUE::Welcome, Road Runner!::Join the Road Runners!]]

- **XML**

<convio:session name="45" param="1001"></convio:session>

S47 TeamRaiser Custom Component

Use the S47 tag to render a custom page component in a TeamRaiser page. In the WYSIWYG editor, the label includes the component custom ID. Displays in the TeamRaiser Event Manager Center preview.

Caution: This component can slow page loading and should be used with great care.

Application

TeamRaiser

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**

```
[[S47:fr_ID:ctag_file:ctag_type:ctag_name]]
```

- **XML**

```
<convio:session name="47" param="ctag_file:ctag_type:ctag_name"/>
```

Parameters

- *fr_ID* - the ID of the TeamRaiser. If it is zero and the S tag is on any non-administrative TeamRaiser page, then the ID is retrieved from the HTTP request.
- *ctag_file* - the base name of a custom page's XML file. For example, "fr_info" would be used to retrieve a component from the "fr_info.xml" file.
- *ctag_type* - one of the values defined in Custom Component Tags.
- *ctag_name* - the identifying text label given to the custom tag.

Sample Code

- **Standard Input**

```
[[S47:1000:fr_info.xml:1:Road Runners]] ([[S47:1000:fr_info:1:event_title]])
```

- **Convio Tag Input**

```
<convio:session name="47" param="1000:fr_info.xml:1:Road Runners"/>
```


Understanding

The Teamraiser pages (such as fr_info.xml in the example above) are stored as XML pages in the Custom Page framework, which in the site data tree on each site. The folder storing the pages for fr_id=1000 on the Support 104 site (which runs on support100) would be this:

/etc/convio/site_data/104/00000104/custom/26/1000/

(where the /104/ is the site id, /26/ is the application id for teamraiser, and /1000/ is the fr_id of the teamraiser object).

Page names within this folder are generally fairly easy to map to the page that you edit in the user interface:

Page in the Admin UI

XML File Name

Greeting	fr_info.xml
Find a Participant	fr_part_finder.xml
Team Selection	fr_team_finder.xml
Team Password	fr_team_password.xml
Participation Options	fr_part_type.xml
Returning Participant Login	fr_user_type.xml
Registration Information	fr_register.xml
Secondary Registration Information	fr_reg_another.xml
Waiver	fr_waiver.xml
Registration Summary	fr_reg_summary.xml
Billing Information	fr_billing.xml
Confirmation	fr_confirm.xml
Thank You page	fr_thanks.xml
Home Page (Participant Center Home Page)	fr_center.xml
Default Personal Page	fr_personal.xml
Company List Page	fr_company_list.xml
Default Company Page	fr_company_page.xml
Team List Page	fr_team_list.xml
Default Team Page	fr_team_page.xml
Top Participants List Page	fr_top_participant_list.xml

Look in the XML file name for the C-Tag that contains the information you want to pull in. For example, the Event Location information in fr_info.xml looks like this:

```
<Component ctag="C1:event_location">
  <Default key="field_size">60</Default>
  <Default key="field_max_length">255</Default>
  <Default key="field_required">false</Default>
  <Shared key="event_location.field"></Shared>
</Component>
```

The ctag_type is the number after the C, in this case "1". To include this for fr_id 1000, the tag would be:

[[S47:1000:fr_info:1:event_location]]

S48 TeamRaiser Participant Information

Use the S48 tag to render summary information for a TeamRaiser participant.

Note: If the value of frID is a Personal Fundraising campaign, use the S48 tag.

Application

TeamRaiser

WYSIWYG Equivalent

Unlike most S tags, S48 tags are added in the WYSISYG editor by clicking the **Convio Conditional** button, rather than clicking the **Convio Component** button.

To add an S48 tag in COM:

- Click **Conditionals** and select **TeamRaiser > All Conditionals**.

To add an S48 tag in CMS:

1. In the CMS WYSIWYG editor, click **Insert Component**.
2. Select **CRM Content** from the Component Type drop-down list.
The COM WYSIWYG editor displays.
3. In the COM WYSIWYG editor, click **Convio Conditional**.
The **Insert Convio Conditional** menu displays.
4. Select **TeamRaiser > All Conditionals** and click **Insert**.
5. Click Finish to close the COM WYSIWYG editor.

CMS Compatible

Yes

Format

- **Standard**

```
[[S48:fr_ID:type:param]]
```

- **XML**

```
<convio:session name="48" param="fr_ID:type:param"/>
```

Parameters

fr_id

the ID of the TeamRaiser

type

one of:

part-type

The name of the participation type

part-type-id

The id of the participation type

team-id

The id of the team or blank if not on a team

team-name

The name of the team

company-name

The name of the company if on a company team

hon-team-id

The honorary team id

hon-team-name

The honorary team name

goal

The participant's goal

dollars

The dollars raised so far by the participant (Note that the full meaning varies depending on SDP settings)

donations

The number of donations raised by the participant (Note that the full meaning varies depending on SDP settings)

emails-sent

The number of e-mails the participant has sent.

pct-of-goal

The percent of goal that the participant has achieved.

goal-remaining

The amount remaining to reach goal.

days-since-reg

The number of days since the participant registered.

days-to-event

The number of days until the event.

race-number

The participant's assigned race number.

participant-id

The participant's cons_id.

fr-id

The TeamRaiser ID.

question

The participant's response to a particular question.

add-on-value

The participant's choice for a particular add-on.

add-on-cost

The participant's cost associated with choosing that add-on.

tent-eligible

Is the participant eligible for tenting (TRUE or FALSE)

tent-status-num

The numeric value of the participant's tenting status.

tent-status-text

The text equivalent (from message catalog) of the participant's tenting status.

tent-confirmation-num

The confirmation number for the participant's tenting assignment.

tent-site

The participant's tent site.

tent-instructions

The participant's tenting instructions.

tentmate-consid

The participant's tentmate's cons_id.

tentmate-name

The participant's tentmate's name.

dollars-online

Money raised online by the participant

dollars-offline-confirmed

Confirmed offline dollars raised by the participant

dollars-offline-unconfirmed

Offline dollars not yet confirmed for the participant

dollars-offline-all

Total offline dollars for the participant

dollars-all-confirmed

Total of all online and offline confirmed dollars raised by the participant

dollars-all

Total dollars raised offline or online, confirmed or unconfirmed regardless of SDP settings

donations-online

See above, but change dollars to number of donations

donations-offline-confirmed

See above, but change dollars to number of donations

donations-offline-unconfirmed

See above, but change dollars to number of donations

donations-offline-all

See above, but change dollars to number of donations

donations-all-confirmed

See above, but change dollars to number of donations

donations-all

See above, but change dollars to number of donations

if-dollars-gt

If dollars raised are greater than the param return TRUE.

if-dollars-lt

If dollars raised are less than the param return TRUE.

if-donations-gt

If donations solicited are greater than the param return TRUE.

if-donations-lt

If donations solicited are less than the param return TRUE.

if-emails-gt

If emails sent by the participant is greater than param, TRUE.

if-emails-lt

If emails sent by the participant is less than param, TRUE.

if-percent-gt

If percent of goal achieved is greater than param, TRUE.

if-percent-lt

If percent of goal achieved is less than param, TRUE.

if-is-captain

If the participant is a team captain, TRUE.

if-on-team

If the participant is on a team, TRUE.

if-page-updated

If the participant has updated his personal page, TRUE.

if-has-participant-center

TRUE if the participant has a participant center (e.g. not a secondary registration)

if-in-group

TRUE if the participant is a member of the group specified. Note that this is different from an S45 which checks the current user for group membership when in the context of someone looking at someone else's personal page or making a donation on someone else's behalf.

if-is-self-donor

If the participant has made a self donation, TRUE. (Support for this type begins in the Aspen release.)

cons.FIELD

Renders the given field for the participant's constituent record

screenname

Renders the screenname for the participant's registration

isAnonymous

Renders true if the registration is anonymous false otherwise

param

a type-specific parameter used thusly:

- for any conditional type, the value to compare against.
- for question, the text of the question that you want to retrieve.

Notes

You can use 0 instead of the fr_id if you are using this on a TeamRaiser to refer to the TeamRaiser that you are on.

This tag will work on a donation form used for TeamRaiser as well as on a TeamRaiser page

If the fr_id corresponds to a Personal Fundraising campaign, this tag will defer to S548 (see that tag for supported types).

Rather than rendering in the context of the logged in constituent, you can pass a specific cons_id like so: `[[S48:fr_id-cons_id:type:param]]`

To test if a logged in constituent is registered for a TeamRaiser event, conditionalize on part-type-id, e.g. `[[?xx::xS48:0:part-type-idx::not registered::registered]]`. You should **not** conditionalize on if-has-participant-center, as this will be FALSE for some registered participants.

Sample Code

- **Standard Input**

```
[[S48:1001001:dollars]]
```

- **Convio Tag Input**

```
<convio:session name="48" param="1001001:dollars"/>
```


S50 HTTP Request Header

Use the S50 tag to render HTTP request header data.

Note: Some parameters may not be available, depending on the user's browser. The W3 provides great detail about the values of the parameters at www.w3.org/Protocols/HTTP/HTTRQ_Headers.html.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S50:parameter]]
- **XML**
<convio:session name="50" param="parameter"/>

Parameters

- *From* - Email address of user
- *Accept* - File types accepted
- *Accept-Encoding* - File encoding accepted
- *Accept-Language* - Human language accepted
- *User-Agent* - Identity of browser
- *Referer* - URI of document from which the current URI was obtained
- *Authorization* - Specifies authorization system and parameters
- *Charge-To* - Specifies system and parameters for billing the end user
- *If-Modified-Since* - Date for comparison with server's data
- *Pragma* - Comma separated list of server-specific directives

Sample Code

- **Standard Input**

```
[[?[[S50:User-Agent]]::MSIE::<link href="../css/UserIEStyle.css"
rel="stylesheet" type="text/css" />::]]
```

- **XML**

```
<convio:session name="50" param="User-Agent"></convio:session>
```

S51 Pagebuilder Content

Use the S51 tag to render content of pages that are in the Page Library. Those pages are by default reusable content. This means that instead of having to use the WYSIWYG actually to insert a page, you type in `[[S51:]]` and the page name.

Application

PageBuilder

WYSIWYG Equivalent

Content > Reusable page

CMS Compatible

Yes

Format

- **Standard**

`[[S51:pagename/category:wrap]]`

- **XML**

`<convio:session name="51" param="pagename/category:wrap"/>`

Parameters

- *pagename* - the name of the page to render. Category is optional. If it is omitted, PageServer's best available algorithm will be used. If a category is specified, the content will fall back to the best available if a page is not available in that category. That may be a PageNavigator-style path such as "topfolder/folder/pagename." Because the slash character was initially used as the separator for the category and is also the path separator for folders, the token after the final slash is interpreted as a category ID if it is numeric but the page name if it is not numeric.
- *wrap* - "true" or "false" and determines whether the content is rendered in the page wrapper. The contents of `[[S51:page:true]]` are identical to `PageServer?pagename=page`. Wrap is optional and defaults to "false."

Sample Code

- **Standard Input**

If your page is called "reus_promoPage"

`[[S51:reus_promoPage]]`

- **XML**

`<convio:session name="51" param="reus_promoPage"></convio:session>`

S52 New Search Component

Use the S52 tag to render a new StoryBuilder search component.

Application

StoryBuilder

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S52:interest]]
- **XML**
<convio:session name="52" param="interest" />

Parameters

interest - an interest category ID that filters search results

Sample Code

- **Standard Input**
[[S52:1234]]
- **XML**
<convio:session name="52" param="1234"></convio:session>

S53 Security Category Prefix

Use the S53 tag to retrieve an abbreviation prefix string for a security category (Multi-Affiliate Single Site mode). There are no security restrictions; this will render even if the user has no permissions for the category. The site data parameter CAT_USE_SECURITY_ABBREVIATION must be set to "TRUE" for this tag to be useful.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**
[[S53:category_ID]]
- **XML**
<convio:session name="53" param="category_ID" />

Parameters

category_ID - ID of the security category

Sample Code

- **Standard Input**
[[S53:5]]
- **XML**
<convio:session name="53" param="5"></convio:session>

S54 Reward Point Information

Use the S54 tag to render reward point information for the current user.

Application

Rewards

WYSIWYG Equivalent

Unlike most S tags, S54 tags are added in the WYSISYG editor by clicking the **Convio Personalization** button, rather than clicking the **Convio Component** button.

Click **Personalization** and select **Rewards > Rewards Point Balance**.

CMS Compatible

Yes

Format

- **Standard**
[[S54:balance]]
- **XML**
<convio:session name="54" param="balance" />

Parameters

balance - Render the user's current balance of points

Sample Code

- **Standard Input**
[[S54:balance]]
- **XML**
<convio:session name="54" param="balance"></convio:session>

S55 Random Number

Use the S55 tag to generate a random number, an integer between a specified minimum and maximum, optionally zero-padded to at least some specified number of digits (000005).

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S55:max]]
[[S55:min,max]]
[[S55:min,max,mindigits]]
- **XML**
<convio:session name="55" param=" min,max,mindigits" />

Parameters

- *min* - the minimum number. Default is 1. Must be between 0 and max, inclusive.
- *max* - the maximum number. Default is 1. Must be between min and 100000000, inclusive. max defaults to 9 if the parameter parse fails.
- *mindigits* - the minimum number of digits for padded output formatting. Default is 1. Must be between 1 and 10, inclusive. For example, if mindigits is 3 and the random number is 4, then the output would be "004".

Sample Code

- **Standard Input**
Display a random image from among a group of JPG files named randomPic1.jpg through randomPic9.jpg.

- **XML**
<convio:session name="55" param="1,9,1"></convio:session>

S56 Find Representatives Component

Use the S56 tag to render the Find Representatives Component.

Application

Advocacy

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S56:V1:0:0:0:1:1:0:0:0:1:1:1:1:0:1:1:0:0:0:0:0:0:0:0:0]]
- **XML**
<convio:session name="56" param="V1:0:0:0:1:1:0:0:0:1:1:1:1:0:1:1:0:0:0:0:0:0:0:0" />

Parameters

With the exception of version, num_national_links, and num_state_links, all values are booleans. If no parameters are set, the tag defaults to

[[S56:V1:1:1:1:1:1:1:1:1:1:1:1:1:1:5:1:1:1:1:1:1]]

version - V1

include_president - 0 or 1

include_vice_president - 0 or 1

include_first_lady - 0 or 1

include_US_senators - 0 or 1

include_US_reps - 0 or 1

include_governors - 0 or 1

include_state_senators - 0 or 1

include_state_reps - 0 or 1

include_postal_address - 0 or 1

include_phone_number - 0 or 1

include_fax_number - 0 or 1

include_email_address - 0 or 1

include_web_site - 0 or 1

include_whitehouse_link - 0 or 1
include_senate_link - 0 or 1
include_house_link - 0 or 1
include_national_links - 0 or 1
num_national_links - Number of links to include under National Issues on results page
include_state_links - 0 or 1
num_state_links - Number of links to include under State Issues on results page
include_county_officials - 0 or 1
include_municipal_officials - 0 or 1
include_facebook_link - 0 or 1
include_twitter_link - 0 or 1
include_youtube_link - 0 or 1
include_blog_officials - 0 or 1
include_committee_assignments - 0 or 1

Sample Code

Show all US senators and US representatives. Display postal address, phone/fax, email and web site for each. Include the senate and house link.

- **Standard Input**
 [[S56:V1:0:0:0:1:1:0:0:0:1:1:1:1:0:1:1:0:0:0:0:0:0:0:0:0:0]]
- **XML**
 <convio:session name="56" param="V1:0:0:0:1:1:0:0:0:1:1:1:1:0:1:1:0:0:0:0:0:0:0:0:0:0" />

S57 Premium Selection Component

Use the S57 tag to render a Premium Selection Component.

Application

Donations

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S57:campaign_ID:donation_level_ID:style:separator or filter]]
- **XML**
<convio:session name="57"
param="campaign_ID:donation_level_ID:style:separator or filter " />

Parameters

- *campaign_ID* - the ID of the campaign to show
- *donation_level_ID* - the ID of a donation level or 0 to show all levels for that campaign
- *style* - 1 for single drop-down, 2 for double drop-down
- *separator or filter* - for a double drop-down, it is the string that separates the group (e.g. show) from the premium. For a single drop-down, it is a filter to use that all premiums must contain in order to be displayed.

Sample Code

Show all premiums available (separator = " - " so that "T-shirt" doesn't get picked up as a group of T and a premium of shirt).

- **Standard Input**
[[S57:1001:0:2:-]]
- **XML**
<convio:session name="57" param="1001:0:2:-"></convio:session>

S58 Email Message

Use the S58 tag to render the text version of the message automatically if embedded in another message or in an autoresponder being rendered in text mode.

To include an email message within the QuickReg or AutoReg autoresponder, just add it to the site data parameter. For example, set the site data parameter CONS_AUTOREG_PARA_2 to `[[S58:1001]]` where "1001" is the ID of the message that you want to include in the automatic registration message from other applications (such as donations).

Application

Communications

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
`[[S58:message_ID:wrap]]`
- **XML**
`<convio:session name="58" param="message_ID:wrap" />`

Parameters

- *message_ID* - the ID of the email message
- *wrap* - "true" to display inside of the email stationery, "false" to display just the message body (default is "false")

Sample Code

- **Standard Input**
`[[S58:1001.0:true]]`
- **XML**
`<convio:session name="58" param=" 1001.0:true " />`

S59 Date String

Use the S59 tag to render a date string in milliseconds.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S59: year/month/day]]
- **XML**
<convio:session name="59" param="year/month/day" />

Parameters

- *year* - four digit year
- *month* - two digit month in which January = 01 and December = 12
- *day* - day as integer with no leading 0 -- the first is 1 not 01

Sample Code

Most commonly used in custom reports to convert string date information into a Convio time stamp.

- **Standard Input**
[[S59:2006/03/3]]
- **XML**
<convio:session name="59" param="2006/03/3"></convio:session>

S61 Facebook Pages

Use the S61 tag to render content for display on Facebook pages. The tag can also return values that may be useful for linking from Convio-hosted pages to Facebook.

For more information on the S61 tag see the [Toolkit for a Convio Application Built on the Facebook Platform](http://open.convio.com/extensions/assets/convio-facebook-application-toolkit.pdf) (<http://open.convio.com/extensions/assets/convio-facebook-application-toolkit.pdf>).

Application

Facebook

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S61:type:parameter1]]
- **XML**
<convio:session name="61" param="type:parameter1"></convio:session>

The full set of bracket tag types is listed below:

- [[S61:logged-in]]
- [[S61:in-canvas]]
- [[S61:in-iframe]]
- [[S61:is-app-added]]
- [[S61:is-viewing-self]]
- [[S61:add-app-url]]
- [[S61:canvas-url]]
- [[S61:user-ID]]
- [[S61:user-count]]
- [[S61:user-table:{public | friends}:{columns}:{max users}]]
- [[S61:referral-count:{any | add_app | advocacy | donation | don_classic | ecommerce | event_guest | event_host | rsvp | teamraiser | ticket | tribute}]]

- [[S61:interaction-count:{any | add_app | advocacy | donation | don_classic | ecommerce | event_guest | event_host | rsvp | teamraiser | ticket | tribute}]]
- [[S61:interaction-list:{public | friends | personal}:{any | add_app | advocacy | donation | don_classic | ecommerce | event_guest | event_host | rsvp | teamraiser | ticket | tribute}:{campaign ID}:{max to show}]]
- [[S61:invitation-form:{invitation type}:{button label}:{any | add_app | advocacy | donation | don_classic | ecommerce | event_guest | event_host | rsvp | teamraiser | ticket | tribute}:{campaign ID}]]
- [[S61:notification-form:{Default Subject}:{Default Text}]]
- [[S61:privacy-form]]
- [[S61:survey:{Survey ID}:{Button Label}]]
- [[S61:gift-form:{Gift Type}:{Radio Label}:{Button Label}:{Default Text}]]
- [[S61:gift-count:{Gift Type}:{Gift Status}]]
- [[S61:gift-log:{Gift Type}:{Gift Status}:{columns}:{max gifts}:{heading}]]
- [[S61:unique-gift-log:{Gift Type}:{Gift Status}:{columns}:{max gifts}:{heading}]]

Parameters

logged-in

returns TRUE if the user is logged in to Facebook (and the Convio Facebook application); otherwise FALSE.

in-canvas

returns TRUE if the content is rendering within a Facebook Canvas page; FALSE otherwise.

Note: this parameter returns FALSE with content rendered within the user's Facebook profile, a notification, etc.

in-iframe

returns TRUE if the content is rendering within a Facebook iframe; FALSE otherwise. We do not currently use the iframe model for our integration (use FBML model instead), so this should always be FALSE.

is-app-added

returns TRUE if the user has added the Convio/Facebook integration to his set of Facebook applications; otherwise FALSE. As the application is normally configured only to show content pages to application users, use this tag in the main wrapper template.

Note: An FBML equivalent to is-app-added should be used for rendering profile content.

is-viewing-self

returns TRUE if the user is viewing his own interactions (in the interaction-list tag); returns FALSE if he is viewing someone else's. The standard configuration of the Convio/Facebook integration allows you to view the interactions of another Facebook user provided you have the argument ID=xxxx in the URL. This tag tells you if that argument is present or not, or if it matches the logged in user.

add-app-url

renders the URL to add the application to your Facebook profile. This is something that you would be likely to use on a regular page displayed on the site or within the profile page.

canvas-url

renders the url `http://apps.facebook.com/S0:FACEBOOK_APP_NAME/`

user-ID

renders the facebook ID of the logged in user. Only marginally useful: all FBML tags that require a user ID will support ID="loggedinuser", and most Facebook URLs that have ID=xxx parameters will assume the logged in user.

user-count

renders the count of users who have added the application.

user-table:{visibility}:{columns}:{max users}

renders a table of users including a thumbnail, name, and profile link.

visibility

Either public or friends (all users versus just the viewer's friends).

columns

How many columns to render.

max users

Maximum number of users (not rows) to render.

referral-count:{interaction type}

renders a count of how many actions of the requested type have occurred as a result of others following links from the user's profile or from an invitation or notification that he sent.

interaction type

One of: any, add_app, advocacy, donation, don_classic, ecommerce, event_guest, event_host, rsvp, teamraiser, ticket, tribute. (Use the add_app interaction type to return the number of new users referred.)

interaction-count:{interaction type}

renders a count of the user's actions of the requested type. Only counts interactions originating on Facebook.

interaction type

One of: any, add_app, advocacy, donation, don_classic, ecommerce, event_guest, event_host, rsvp, teamraiser, ticket, tribute.

interaction-list:{visibility}:{interaction type}:{campaign ID}:{max to show}

renders a list of all the defined (by visibility) group's actions of the requested type. The standard rendering includes a picture of the user who took action along with links to that user's profile and to the action. Only counts interactions originating on Facebook. The interactions are ordered from most recent to oldest.

visibility

One of public, friends or personal indicating all users, just viewer's friends, or just the user.

interaction type

One of: add_app, advocacy, donation, don_classic, ecommerce, event_guest, event_host, rsvp, teamraiser, ticket, tribute.

campaign ID

A non-zero value scoping the interactions to a specific campaign, donation form, action alert, store, or TeamRaiser.

max to show

Maximum number of interactions to render.

invitation-form:{invitation type}:{button label}:{interaction type}:{campaign ID}

renders an invitation to friends to take an action specified by the interaction type and campaign ID.

Note: Invitations and invitation forms have limited functionality in Facebook.

invitation type

The recipient sees "You have a __ invitation;" therefore, such values as the organization's name, or "fundraising" or "advocacy" make sense.

button label

The button shows below the invitation text and the sender's picture. Such values as "Take Action Now" and "Add this way cool application" make sense (depending on the target).

interaction type

One of: add_app, advocacy, donation, don_classic, ecommerce, event_guest, event_host, rsvp, teamraiser, ticket, tribute.

campaign ID

The non-zero value to direct the recipient to a specific campaign, donation form, action alert, store, or TeamRaiser. If zero, the invitation goes to the page to add the application. If the interaction type is add-app, then this value is irrelevant.

notification-form:{Default Subject}:{Default Text}

renders an interface similar to Tell-a-friend for sending short messages to up to 10 friends.

Default Subject

The value for the form subject box. The user can override this value.

Default Text

Notifications always start with the sender's name, so this text should open with a verb. For example, "**supports** the site." would make sense. This text will always render first in the notification, but the user can append additional content.

privacy-form

renders the form to update the privacy options within Convio.

survey:{Survey ID}:{Button Label}

embeds a survey within a Facebook canvas page. As this honors the NEXTURL survey feature, explicitly redirect the user to another page after completion. If no NEXTURL is provided, then the same page will load replacing the survey with the responses in read-only format.

Survey ID

The ID of the survey to display.

Button Label

The submit button label.

gift-form:{Gift Type}:{Radio Label}:{Button Label}:{Default Text}

renders a form for sending gifts to your friends or yourself.

Gift Type

Filter gifts shown to a particular type. If blank, all gift types will show.

Radio Label

The label of the radio button clicked to select a gift. Values like "Select this gift" or "Adopt/Send" would make sense.

Button Label

The label for the submit button.

Default Text

Notifications always start with the sender's name, so this text should open with a verb. For example, "**sent** you a gift." would make sense. The user can append additional text after this opening text, which always shows.

gift-count:{Gift Type}:{Gift Status}

returns the total number of gifts received by the user including gifts to self and those marked private. Gifts deleted by the user or by the administrator are not included.

Gift Type

Filter gifts shown to a particular type. If blank, all gift types will show.

Gift Status

Filter the set of gifts shown to available or unavailable. If blank all status values display.

gift-log:{Gift Type}:{Gift Status}:{columns}:{max gifts}:{heading}

Returns the total number of gifts received by the user including gifts to self. Private gifts only display outside the profile page when viewed by the recipient. The order of display is most recent to oldest.

Gift Type

Filter gifts shown to a particular type. If blank, all gift types will show.

Gift Status

Filter the set of gifts shown to available or unavailable. If blank all status values display.

columns

How many columns the table has. Default is 1. One column display is special, showing the gift message text, gift date, and links to make the gift private or to delete it.

max gifts

Maximum number of gifts (not rows) in the table.

heading

Heading to display above the table of gifts.

unique-gift-log:{Gift Type}:{Gift Status}:{columns}:{max gifts}:{heading}

Renders a list of gifts received by the user including gifts to self. Private gifts only display outside the profile page when viewed by the recipient. This differs from a gift-list in that only the most recent instance of particular gift is rendered. Possible duplicate gifts and space limitations make this a good choice for profile pages.

Gift Type

Filter gifts shown to a particular type. If blank, all gift types will show.

Gift Status

Filter the set of gifts shown to available or unavailable. If blank all status values display.

columns

How many columns the table has. Default is 1. One column display is special, showing the gift message text, gift date, and links to make the gift private or to delete it.

max gifts

Maximum number of gifts (not rows) in the table.

heading

Heading to display above the table of gifts.

Sample Code

- **Standard Input**
[[S61:referral-count:any]]
- **XML**
<convio:session name="61" param="referral-count:any"></convio:session>

S63 Lorem Ipsum Sample Text

Use the S63 tag to render Adobe's Lorem Ipsum Sample Text. Specify the specified number of paragraphs (from 1 to 3) of the standard "Lorem ipsum dolor..." for preview of a design. This is a placeholder for use in PageWrappers and Stationery. The space occupied by the sample text is reserved for later replacement with real content.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S63:num_paragraphs]]
- **XML**
<convio:session name="63" param="num_paragraphs" />

Parameters

num_paragraphs - the number of paragraphs, from one to three, to render

Sample Code

- **Standard Input**
[[S63:2]]
- **XML**
<convio:session name="63" param="2" />

S65 Directory Quick Search Component

Use the S65 tag to display a Quick Search component for the directory application.

Application

Directory

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**
[[S65: field_width:titled]]
- **XML**
<convio:session name="65" param=" field_width:title" />

Parameters

field_width - the width of the text input field

title - a title to display above the input field

Sample Code

- **Standard Input**
[[S65:50:Search Your Directories]]
- **XML**
<convio:session name="65" param="50:Search Your Directories">
</convio:session>

S66 Embed Another Template

Use the S66 tag to embed and render another template.

S66 retrieves the named template from the appropriate template set cache and renders it using the same render context ("A" tags) as the calling template.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**
[[S66:template_filename]]
- **XML**
<convio:session name="66" param="template_filename"/>

Parameters

template_filename - The name of the template file relative to the current template set. Allowed characters are ".", "/", "_", "-", and alpha-numerics.

Sample Code

- **Standard Input**
[[S66:template_macros/Use_Me_Everywhere.tpt]]
- **XML**
<convio:session name="66" param="template_macros/Use_Me_Everywhere.tpt"></convio:session>

S67 Qualified Contributions

Use the S67 tag to render qualified contributed content.

Application

Solicited Content

WYSIWYG Equivalent

Solicited Content > Qualified Contribution

CMS Compatible

Yes

Format

- **Standard**

[[S67:itemtype:criteria:altitem:layout:creation:exists]]

- **XML**

<convio:session name="67" param=="itemtype:criteria:altitem:layout:creation_date:exists"/>

Parameters

- *itemtype* - The Solicitation ID of the qualified content.
- *criteria* - Filtering criteria. Values include *country*, *state*, *security_category*, and *affiliate*.
- *altitem* - The ID of an alternative contribution to display if the qualified content cannot be found.
- *layout* - Specifies how the qualified contributed content is formatted. A layout specified in the XML file is located here:

site_data/001/00000001/custom/50/regional_insert_format_*.xml

For example:

site_data/001/00000001/custom/50/regional_insert_format_news.xml

- *creation date* - Contributions checks against this date to determine if the content qualifies.
- *exists* - A Boolean check.
 - If set to *true*, the S tag renders the value *TRUE* or *FALSE* rather than the content.
 - If set to *false*, the content is rendered rather than the Boolean check.

This allows you to test that the content renders before it displays.

Sample Code

- **Standard Input**
`[[S67:240:affiliate:0:standard:6/26/2009:true]]`
- **XML**
`<convio:session name="67"
param="240:affiliate:0:standard:6/26/2009:true"></convio:session>`

S68 Specific Contributions

This component renders a specific contribution. If you do not specify a layout parameter, the default layout is used.

Application

Solicited Content

WYSIWYG Equivalent

Solicited Content > Specific Contribution

CMS Compatible

Yes

Format

- **Standard**

[[S68:itemid:layout]]

- **XML**

<convio:session name="68" param=="itemid:layout"/>

Parameters

- *itemid* - The specific contribution ID.
- *layout* - How the content is formatted.

Sample Code

- **Standard Input**

[[S68:1001:layout1a]]

- **XML**

<convio:session name="67" param="1001:layout1a"></convio:session>

S69 Time Contributions

Use the S69 tag to render time-constrained contributed content.

Application

Solicited Content

WYSIWYG Equivalent

Solicited Content > Time-Based Contribution

CMS Compatible

Yes

Format

- **Standard**

[[S69:itemtypeid:layout:abort]]

- **XML**

<convio:session name="69" param==" itemtypeid:layout:abort" />

Parameters

- *itemtypeid* - The ID of the Solicitation being used.
- *layout* - How the contribution be formatted.
- *abort* Specifies what to do if no valid contributions are found for the solicitation chosen.

Contributions that have publish and expire dates are evaluated at render time. Contributions are valid if the current time is past their publish time but before their expire time.

If the abort boolean is true, no email will be sent if there is no valid contribution.

If the abort boolean is false, a blank email is sent if there is no valid contribution.

Sample Code

- **Standard Input**

[[S69:1001:layout1b:true]]

- **XML**

<convio:session name="69" param=" 1001:layout1b:true"></convio:session>

S72 Localization (L10N) Information

Use the S72 tag to render data in a locale-specific format, when appropriate. This is especially useful for currency.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S72: info_type:parameter1:parameter2]]
- **XML**
<convio:session name="72" param=" info_type:parameter1:parameter2 " />

Parameters

info_type – The *infotype* parameter can include any of the following options:

- *locale* - Return the locale in `Locale.toString()` format (e.g. "en_US, es, fr_CA").
- *currency_locale* - Returns the currency locale in `Locale.toString()` format (e.g. "en_GB").
- *currency_name* - Returns the name of the currency corresponding to the currency locale, in the style specified by parameter1. A typical style would be "singular". The name is determined by looking in the "currency" message catalog for a message that has the key "currency code.name.parameter1" where currency code is the currency code (e.g. "USD") corresponding to the currency locale. For example, if the contents of the "currency" message catalog was as follows:

Key	Value
USD.name.singular	dollar
USD.name.plural	dollars
USD.name.singular_capitalized	Dollar
USD.name.plural_capitalized	Dollars

GBP.name.singular	pound
GBP.name.plural	pounds
GBP.name.singular_capitalized	Pound
GBP.name.plural_capitalized	Pounds

and the currency locale was "en_GB", then [[S72:currency_name:plural]] would return "pounds".

- *is_locale_enabled*
Return "true" or "false" depending upon whether the locale specified by parameter1 is enabled for the site. parameter1 must be specified in Locale.toString() format (e.g. "fr_CA"). For example, if the "en_GB" locale is enabled for the site, then [[S72:is_locale_enabled:en_GB]] will return "true".
- *currency_symbol*
Returns the currency symbol (e.g. "\$", "£", etc.).
- *currency_code*
Returns the currency code (e.g. "USD", "GBP", etc.).
- *format_currency*
Formats parameter1 as a currency value in a manner that is appropriate for the currency locale. parameter1 specifies the value in the currency's non-fractional unit (e.g. cents for "en_US", pence for "en_GB").
- *format_currency_zero*
Shorthand for [[S27:format_currency:0]].
- *short_name*
Returns the short display name for the locale specified by parameter1 rendered in the locale specified by parameter2. For example, if both parameter1 and parameter2 were specified as "fr_CA", then this tag would return "français (CA)".

Sample Code

The following will return "\$29.99" in the "en_US" locale:

- **Standard Input**
[[S72:format_currency:2999]].
- **XML**
<convio:session name="72"
param="format_currency:2999"></convio:session>

S75 Vote Information

Use the S75 tag to render information about how representatives vote on an issue.

Application

Advocacy

WYSIWYG Equivalent

Advocacy > Vote Info

CMS Compatible

Yes

Format

Standard

[[S75:id:vote id:option:param]]

Parameters

- *Title* - Vote title
- *Rollcall_number* - Roll call number
- *Description* - Vote description
- *Rollcall_description* - Description of the roll call.
- *Date* - Vote date
- *Congress_session_info* - Congress and session
- *Result* - Vote result
- *Tally* - Vote tally
- *Position* - Organization's position

Sample Code

- **Standard Input**
[[S75:101:title]]
- **XML**
<convio:session name="75" param="101:title"></convio:session>

S79 Specific Solicited Content

Use the S79 tag to render the correct contribution from a specific solicitation in contributed content.

Application

Solicited Content

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**

[[S79:itemtype:criteria:criteriavalue:altitem: layout]]

- **XML**

<convio:session name="79" param=" info_type:parameter1:parameter2 " />

Parameters

- *itemtypeid* - The ID of the criteria used to generate the solicitation. Values include *affiliate*, *country*, *state*, and *security_category*
- *criteriavalue* - A valid affiliate code, country value, state value, or security category
- *altitem* - The ID of an alternate contribution to render if no matching contributions can be found for this solicitation
- *layout* - Desired layout for the contribution. If this is blank, the default will be used.

Sample Code

- **Standard Input**

[[S79:1001:state:TX:1011:layout1c]]

- **XML**

<convio:session name="79" param="1001:state:TX:1011:layout1c"></convio:session>

S80 Value of Session Variable

Use the S80 tag to render a session variable's value. Session variables are typically set by the user with `[[U0]]`, `[[U1]]`, or in a URL parameter by prefixing "s_" to the variable name.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
`[[S80:variable_name]]`
- **XML**
`<convio:session name="80" param="variable_name" />`

Parameters

variable_name - The name of the session variable to render. If it does not exist, then an empty string is rendered.

Sample Code

- **Standard Input**
To get the current TeamRaiser ID:
`[[S80:trID]]`
To get the ID of the current Security Category:
`[[?xxnullx1x::x[[S80:AffiliateSecCatId]]x::
 (The General Security Category either has no ID# or ID# is 1)
::
 [[S80:AffiliateSecCatId]]
]]`
- **XML**
`<convio:session name="80" param="src">`

S81 Sign-In Component

Use the S81 tag to render a sign-in link. When the user clicks the link it displays the user-side login page.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S81:link_text:title]]
- **XML**
<convio:session name="81" param=" link_text:title" />

Parameters

link_text - title

Sample Code

- **Standard Input**
Already a member? [[S81]]
- **XML**
Already a member? <convio:session name="81"/>

S83 Information About The Current User

Use the S83 tag to render several different types of information about the current center.

Application

Centers

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S83:parameter]]
- **XML**
<convio:session name="83" param="parameter" />

Parameters

- *center_name* - Renders the name of the current center
- *center_ID* - Renders the ID of the current center
- *center_category_ID* - Renders the ID of the security category associated with the current center

Sample Code

- **Standard Input**
You are currently in the [[S83:center_name]] Center.
- **XML**
You are currently in the <convio:session name="83" param="center_name" /> Center.

S87 Display String Literals

Use the S87 tag to display string literals that would otherwise be escaped by the WYSIWYG editor. This includes >, <, and & characters. Useful within other tags.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S87:parameter]]
- **XML**
<convio:session name="87" param="parameter" />

Parameters

parameter - The literal to render. Accepts greater_than, less_than or ampersand.

Sample Code

- **Standard Input**
Spacely Sprockets [[S87:ampersand]] Co.
- **XML**
If x <convio:session name="87" param="less_than" /> 9

S88 Dynamic Ask Component Amount

Use the S88 tag to render a monetary sum returned by a Dynamic Ask Component.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**

[[S88:starting_amount,default_amount,modify_percentage,increment_amount,round_to_nearest]]

- **XML**

```
<convio:session name="88" param="largest_trans_amount,$5.00,-85,$0.00,$1.00 " />
```

Parameters

- *starting_amount* - A monetary amount that can be pulled from the constituent's record via an S1 tag. Using the WYSIWYG, this presents transaction information such as "largest_gift_amount." If a site has a custom monetary field enabled you can use that instead, such as "custom_monetary_amount3."
- *default_amount* - A monetary amount that will act as a default. In theory, it is also a minimum, but outside a donation form level this may not work as a minimum if using a negative percentage in *modify_percentage*. See ENG 23880.
- *modify_percentage* - A percentage to modify the *starting_amount* by. May be positive to increase or negative to decrease the amount.
- *increment_amount* - After applying the *modify_percentage* to *starting_amount*, increment the value by this amount.
- *round_to_nearest* - Round the amount to the nearest multiple of this value.

Sample Code

- **Standard Input**

[[S88:largest_trans_amount,\$5.00,85,\$0.00,\$1.00]]

1. Take the constituent's largest previous transaction or \$5.00 (whichever is greater). For this example, assume a largest previous transaction of \$100.
2. Increment that amount by -85%: $100 + (100 * -0.85) \rightarrow 100 + (-85) \rightarrow 100 - 85 = \15.00
3. Add \$0.00: $15 + 0 = \$15.00$
4. Round to the nearest multiple of \$1.00 = \$15.00.
5. [[S88:custom_monetary_value3,\$10.00,10,\$7.00,\$5.00]]
6. Assume that the site has turned on custom_monetary_value3 and that it holds \$50.00 for this constituent:
 $50 + (50 * 0.1) = 55$
 $55 + 7 = 62$
7. Round to nearest multiple of 5 = \$60

- **XML**

If x <convio:session name="89" param="largest_trans_amount,\$5.00,85,\$0.00,\$1.00" />

S90 Advocacy Alert Information

Use the S90 tag to render action alert information. Does not work in email.

Application

Advocacy

WYSIWYG Equivalent

Advocacy > Alert Info

CMS Compatible

Yes

Format

- **Standard**

[[S90:alert id:param]]

- **XML**

<convio:session name="90" param="largest alert id:param" />

Parameters

- *can_reply_no_default* - Can constituent take action by reply? Renders TRUE or FALSE.
- *Message* - Action alert message
- *Recipient_list* - Action alert targets

Sample Code

- **Standard Input**

[[S90:101:recipient_list]]

- **XML**

If x <convio:session name="90" param=" 101:recipient_list" />

S93 User's Representatives

Use the S93 tag to display the user's elected representatives. The configuration parameters specify the types of representatives to display and offer formatting options. Use in email.

Application

Advocacy

WYSIWYG Equivalent

Advocacy > Your Representative

CMS Compatible

Yes

Format

- **Standard**
[[93:rep_type:output_mode:format_options]]
- **XML**
param="rep_type:output_mode:format_options" />

Parameters

- ***rep_type***

Specifies the type of representative to display. If the representative(s) cannot be determined from the user's address, then default values are provided which are defined in the Advocacy message catalog. Choices are:

rep_type	Representative	Default text	Message catalog key for default text
president	the President	the President	your_reps_default_pres
vice president	the Vice President	the Vice President	your_reps_default_vice_pres
first lady	the First Lady	the First Lady	your_reps_default_first_lady
federal senator	Federal senators	your U.S. senators	your_reps_default_fed_sen
federal representative	Federal representatives	your U.S. representative	your_reps_default_fed_rep
state senator	State senator	your state senator	your_reps_default_state_sen
state representative	State representative	your state representative	your_reps_default_state_rep
governor	State governor	your state governor	your_reps_default_governor

- ***output_mode***

Specifies the basic layout of the representatives. Choices are:

text

Plain text in-line with surrounding content. When there are multiple representatives to display, they are separated with commas except for the last two, which are separated with the word "and" (for federal senators) or "or" (for all other types).

List

Listed one per line.

bulleted

A bulleted list.

raw

Outputs only the value of the fields selected, with no separators between fields or representatives. This is meant for use in conditionals, but may also be used when a single field is desired without the representative name appearing.

- ***format_options***

Specifies which information about the representatives to display. This value is the concatenation of one or more of the following keywords with each separated by underscore characters. The formatting of the specified fields is defined by advocacy/ad_your_rep_row.tpt template file and message catalog entries. The keywords are:

name

The full name of the representative. This includes title, first, middle, last, and suffix. This is the default, if first or last is not given and the style is not **raw**.

first

The first name of the representative. This is overridden if **name** is also specified.

last

The last name of the representative. This is overridden if **name** is also specified.

prefix

The office title of the representative (e.g., "Senator"). The "s93_prefix_name" message catalog entry is used to combine the prefix and the name.

phone

The phone number in their seat of government. For example, for U.S. representatives and senators, the phone number of their Washington D.C. office. The format of the phone number is specified in the "advocacy" message catalog; if the phone number has 10 digits, then the "your_reps_phone_fmt_area" message is used, if it has 7 digits, then the

"your_reps_phone_fmt_no_area" message is used, otherwise, the phone number is rendered as defined in the database without any additional formatting.

distphone

The phone number in their district office. This may only be valuable for members of Congress. If both phone numbers are specified, then the "s93_phones" message catalog entry is used to combine them.

webform

Link to the representative's webform. Use your_reps_webform_link_text to change the content of the hyperlink.

twitter

Link to the representative's twitter page. Use your_reps_twitter_link_text to change the content of the hyperlink.

facebook

Link to the representative's facebook page. Use your_reps_facebook_link_text to change the content of the hyperlink.

blog

Link to the representative's blog. Use your_reps_blog_link_text to change the content of the hyperlink.

youtube

Link to the representative's youtube page. Use your_reps_youtube_link_text to change the content of the hyperlink.

comments

Displays comments about the representative (as entered on the **Additional Information** step of the **Legislator Fact Sheet** page).

WYSIWYG Integration

The WYSIWYG UI presents a fixed set of combinations of the formats specified in the "format_options" parameter. It allows the user to choose one name format and one phone number format (the selected keyword strings are concatenated). Of course, manually entering an "S93" tag allows you to make any other combination.

The name choices are (parentheses indicate the resulting keyword selection and are not shown in the UI):

- Full Name (name)
- First and Last Name (first_last)

- Title and Full Name (prefix_name)
- Title and First and Last Name (prefix_first_last)
- Title and Last Name (prefix_last)

The phone number choices are:

- No Phone
- Primary Phone (phone)
- District Phone (distphone)
- Both (phone_distphone)

Sample Code

- **Standard Input**
[[S93:president:bulleted:prefix_first_last_]]
- **XML**
<convio:session name="93"
param="president:bulleted:prefix_first_last_"></convio:session>

S94 Action Alert List

Use the S94 tag to render a list of action alerts.

Application

Advocacy

WYSIWYG Equivalent

Advocacy > Alert List

CMS Compatible

Yes

Format

- **Standard**
[[S94:region:numLinks:issue:states:taken:desc:sortBy:link:listStyle]]
- **XML**
<convio:session name="94"
param="region:numLinks:issue:states:taken:desc:sortBy:link:listStyle" />

Parameters

Display alerts targeted within this region. Possible values:

- *allalerts* (displays all alerts)
- *consstate* (displays alerts from the constituent's state. User must be logged in and have a known state for this to work)
- *allstates* (display all alerts from all states)
- *national* (display all national alerts (not state affiliated))
- *XX* (where XX is the 2 letter state abbreviation: displays alerts from this state)

How many alerts should be in the list? Any number from 1-999999

Limit the alerts to ones associated with this issue.

To find the issue ID

1. Go to **Advocacy > Issues**.
2. Select an issue and click **Edit Issue**.
3. Look in the URL for "folder_ID=" - that will be the issue ID. Typically this is a 4-digit number such as 1000.

To set what content displays to a user who is not logged in, use these values:

- *allstates* (alerts from all states)
- *fedonly* (only federal alerts)

To set whether an alert displays for users who have already taken the alert (user must be logged in), use these values:

- *show*
- *hide*

To display the Alert Description in the list use these values:

- *description*
- *no*

To set how the list sorts, use these values:

- *priority* (alert priority)
- *expire* (alert expiration date)
- *publish* (alert publish date)

The alert link can be either the alert title, or a separate "take action" link to the right of the title.

To set how the link appears, use these values:

- *title*
- *separate*

To set how the list is formatted use these values:

- *plain*
- *bullet*
- *number*

Sample Code

- **Standard Input**

```
[[S94:region:allstates:numLinks:999999:issue:-
1:states:allstates:taken:show:desc:description:sortBy:priority:link:title:listStyle:
plain]]
```

Displays all state alerts for all issues whether the user is logged in or not, display the description of the alert with the alert title as the link to the alert, and sort the list by priority with no bullets or numbers.

```
[[S94: TX:1:1001:allstates:hide:no:expire:separate:bullet]]
```

Displays 1 alert from Texas associated with Advocacy Interest 1001, but if a user has already taken an alert don't include it, do not display the description, have the link be a separate link, sort the list by alert expiration date, and display the list as a bulleted list.

- **XML**

```
<convio:session name="94" param="allstates:999999:-  
1:allstates:show:description:priority:title:plain "></convio:session>
```

S96 Alerts Taken List Component

Use the S96 tag to render an Alerts Taken List component.

Application

Advocacy

WYSIWYG Equivalent

Advocacy > Alert Taken List

CMS Compatible

Yes

Format

- **Standard**
[[S96: parameters]]
- **XML**
<convio:session name="96" param=" parameters " />

Parameters

Unlike most "S" tag components, the parameters for S96 are a series of *name:value* pairs. The name:value pairs are separated by colons (e.g., "name1:value1:name2:value2:...").

- *region* - Specifies whether to constrain the list of alerts to a particular region. The potential values are:
 - *allalerts*
No constraint.
 - *national*
Only show alerts with no state restrictions.
 - *allstates*
Only show alerts with state restrictions.
 - *consstate*
Only show alerts with a state restriction that matches the constituent's state. If the constituent is not logged in, this is equivalent to "allstates".
 - *stateprov*
Here, stateprov specifies a particular state/province (e.g. "TX"). Only show alerts with a state restriction that matches stateprov. In all cases, if the constituent is logged in, if the alert has any state restrictions, the alert is displayed only if the constituent's home state/province matches.
- *numlinks* - Specifies the maximum number of alerts to display.

- *issue* - In the case where the list should only include alerts that are associated with a particular advocacy issue, specifies the ID of the folder corresponding to an advocacy issue. "-1" indicates there is no constraint.
- *desc* - Specifies whether to display the alert description. The potential values are:
 - *description*
Display the alert description.
 - *no*
Don't display the alert description.
- *date* - Specifies whether to display the date when the constituent took action. The potential values are:
 - *actiondate*
Display the date the constituent took action.
 - *no*
Don't display the date the constituent took action.
- *stats* - Specifies whether to display statistics for the alert. The potential values are:
 - *statistics*
Display statistics for the alert.
 - *no*
Don't display statistics for the alert.
- *tafLink* - Specifies whether to display a Tell-a-Friend link. The potential values are:
 - *none*
Don't display a Tell-a-Friend link.
 - *title*
Make the alert title a Tell-a-Friend link.
 - *separate*
Display the Tell-a-Friend link separately.
- *sortBy* - Specifies how to sort the list. The potential values are:
 - *priority*
Sort by alert priority.
 - *expire*
Sort by alert expiration date.
 - *publish*
Sort by alert publish date.
 - *action*
Sort by the date the constituent took action.

- *listStyle* - Specifies the style in which the list should be rendered. The potential values are:
 - *plain*
Display the list without any bullets or numbers.
 - *bullet*
Display list as a bullet list.
 - *number*
Display list as a numbered list.

Sample Code

- **Standard Input**

[[S96:allstates:999999:-1:allstates:show:description:priority:title:plain]]

Displays all state alerts for all issues whether the user is logged in or not, the description of the alert with the alert title as the link to the alert, and sorts the list by priority with no bullets or numbers.

[[S96:TX:1:1001:allstates:hide:no:expire:separate:bullet]]

Displays 1 alert from Texas associated with Advocacy Interest 1001, but if a user has already taken an alert don't include it, do not display the description, have the link be a separate link, sort the list by alert expiration date, and display the list as a bulleted list.

- **XML**

```
<convio:session name="96" param="allstates:999999:-1:allstates:show:description:priority:title:plain"></convio:session>
```

S97 RSS Link Component

Use the S97 tag to render the RSS link component.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S97:param:param]]
- **XML**
<convio:session name="97" param="param:param" />

Parameters

- *topic* - Specifies which RSS topic area is pointed to by the RSS link. You can run the class `FeedProviderRegistry` to get a list of feed topics.
- *imageFile* - Specifies the image file to use for the link image, relative to `.../static_data/images`. If *imageFile* is "addToMyYahoo", the "Add to My Yahoo!" button is rendered.
- *format* - Specifies the format of the feed that is pointed to by the RSS link. Right now, the only supported formats are "rss2_0" and "rss1_0".
- *rawUrl* - If TRUE, instead of rendering the link to RSS feed within the specified button, only a link is rendered.

Sample Code

- **Standard Input**
[[S97:alerts:feed-rss.gif:rss2_0:false]]
- **XML**
<convio:session name="97" param="alerts:feed-rss.gif:rss2_0:false" />

S98 Time After/Time Before a Given Date

Use the S98 tag to render the number of days, weeks, months or years between a given date and the current date.

Application

Donations2

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S98:parameters]]
- **XML**
<convio:session name="98" param="parameters" />

Parameters

Tip: [[S98:help]] prints a summary of valid parameters.

- *units* - valid parameters are "days", "weeks", "months", and "years"
- *aspect* - valid parameters are "until" and "after"
- *date* - a date literal, which can be produced by an S tag

The most common valid values to format the date literal are:

e	day of week	number	2
e	day of week	text	Tuesday; Tue
y	year	year	1996
d	day of year	number	189
m	month of year	month	July; Jul; 07
d	day of month	number	10
a	halfday	text	PM
k	hour (0~11)	number	0

h	clockhour (1~12)	number	12
h	hour (0~23)	number	0
k	clockhour (1~24)	number	24
m	minute of hour	number	30
s	second of minute	number	55
s	fraction of second	number	978
z	time zone	text	Pacific Standard Time; PST
z	time zone offset/ID	zone	-0800; -08:00; America/Los_Angeles

Sample Code

- **S Tags Input**

To display shopping days remaining until Christmas 2012:

```
[[S98:days:until:2012-12-25:YYYY-MM-dd]]
```

To display days until a constituent's membership expires:

```
[[E98:days:until:{{S1:membership_expiration_date}}:MM DD, YYYY]]
```

- **XML**

```
<convio:session name="98" param="days:until:2012-12-25:YYYY-MM-dd" />
```

S99 URL Shortcut

Use the S99 tag to render an URL shortcut.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S99:key]]
- **XML**
<convio:session name="99" param="key" />

Parameters

key -unique, identifying key for the URL. Keys are as per PageBuilder names; limited to alpha-numeric, "_" (underscore), and "-" (dash) and are not case sensitive.

Sample Code

- **Standard Input**
[[S99:_homepage]]
- **XML**
<convio:session name="99" param="_homepage"]]

S100 Site Data Parameter as Template

Use the S100 tag to apply template tags within site data parameter (SDP) values. The parameter is an SDP name. The value of that SDP is compiled as a template, and then evaluated in the current context. The full expressiveness of the template engine is available within the SDP value, and "A" tag values are passed through from the parent template containing the "S100".

For efficiency, the SDP values are only compiled by the template system once upon first invocation, and compiled Template objects held in the SiteInfo cache. This should be invisible to the user and should not affect the semantics.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**
[[S100:SDP_NAME]]
- **XML**
<convio:session name="100" param="SDP_NAME" />

Parameters

SDP_NAME - the name of the SDP to render

Sample Code

- **Standard Input**
[[S100:_homepage]]
- **XML**
<convio:session name="100" param="_homepage"]]

S101 Days Until Password Expiration

Use the S101 tag to render the number of days until password expiration for the session user. If the expiration date is within the SDP setting for the warning message, the value will be rendered; otherwise, "" (blank) is rendered. If the expiration date is in the past, a negative number for the days past expiration will be rendered.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**
[[S101]]
- **XML**
<convio:session name="101" param="" />

Parameters

None

Sample Code

- **Standard Input**
[[S101]]
- **XML**
<convio:session name="101" param="">]]

S102 Renders a Visual Alert (Warning) Box

Use the S102 tag to pop up an alert box.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S102:message]]
- **XML**
<convio:session name="102" param="message" />

Parameters

message - The message to display in the alert box

Sample Code

- **Standard Input**
[[S102:Your password is about to expire.]]
- **XML**
<convio:session name="102" param="Your password is about to expire."/>

S103 Constituent Attribute Selection List

Use the S103 tag to create an HTML select control listing all values of a constituent record attribute. The tag can set the initially-selected item based on the value of an HTTP parameter. This is especially helpful in donation forms (for example, *set.BillingState*).

For scripting purposes, use the alternate JSON representation.

Application

Reports

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**

[[S103:field:options]]

- **XML**

<convio:session name="" param="field:options"/>

Parameters

- *Field* - The name of the field for which to list the possible values. This is the constituent attribute ID for the field. There are a few aliases that can be used for readability: *country* and *state* can be used in place of *home_country* and *home_stateprov*; the *cons_* prefix can be omitted (e.g., *gender* instead of *cons_gender*).
- *options* - A comma-separated list of optional configuration in the format "option1=value1, option2=value2, ...". Options are:
- *name* - The name for the HTML select list control. The default value is the "field" parameter value.
- *ID* - The ID for the HTML select list control. The default value is the "name" option value.
- *arg* - The name of an HTTP parameter that can set the initially selected list item
- *format* - The output format, either "html" or "json". The default value is "html".

Sample Code

Standard Input

- State HTML selection list in a donation form using Convio API:
S103:state:id=billing_address_state, name=billing.address.state, arg=set.BillingState
- State list for JavaScript to populate a list:
var stateInfo = [[S103:state:arg=set.BillingState, format=json]];
var list = document.getElementById('stateListId');

for (var s = 0; (s < stateInfo.items.length); s++) {
 var state = stateInfo.items[s];
 var opt = document.createElement('OPTION');
 opt.value = state.value;
 opt.appendChild (document.createTextNode (state.label));
 opt.selected = state.selected;
 list.options.add (opt);
}

XML

```
<convio:session name="103" param="state:arg=set.BillingState,  
format=json"/>
```


S120 Object Property

Use the S120 tag to return the string value of a named property from a named object. The object must be stored using `PropertySTagComponent.setObject()`.

The S120 doesn't work in all autoresponders -- you may want to access the `AutoResponderTrick` to verify that the tags work. You also can pull this information in individually.

A comprehensive list of properties available to the S120 tag is listed in Appendix A: S120 Object Properties.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S120:objectName:propertyName[:propertyParameter...]]]
- **XML**
<convio:session name="120" param="objectName:propertyName[:propertyParameter...]" />

Parameters

A property name must correspond to a public accessor method named *getProperty* or *isProperty*. For example, in order to get the value of a "foo" property, there must be a public method with one of the following signatures:

- public type `getFoo(params)`
- public type `isFoo(params)`

Where the parameters are:

- empty (e.g., public `String getFoo()`)
- one or more strings
- a `CVSession` and a `TemplateContext`
- a `CVSession` and a `TemplateContext` followed by one or more strings

In the case of the methods that take a `CVSession` and a `TemplateContext`, the `CVSession` and `TemplateContext` supplied to the call to `render()` will be passed

into the accessor method. The values for any String arguments are specified using "property parameters"; see below. By convention, property names always begin with a lowercase letter. For example, a property returned by the accessor `getFooBar()` would be named "fooBar". As a convenience, the full name of the method can also be used to name a property. This is mainly provided so that tags like `[[S120:dc:isTribute]]` will work.

propertyParameter - Value of parameter to property method Specifies a string value that should be supplied as a String argument value when calling the property method. Multiple property parameters can be specified as required.

renderMode - The tag `[[S120:rc:renderMode]]` will return one of three values: "email", "web", or "printer". (If you are not in the context of a donation, then "rc:renderMode" is returned.) You can use this to customize the autoresponder message depending upon whether it is being rendered in the context of thank you email, the normal web page, or the printer-friendly web page.

Sample Code

Standard Input

- Dear `[[S120:dc:name]]`
`[[?xxnullxx::x[[S120:dc:jointDonorName]]x::: and
[[S120:dc:jointDonorName]]]]`
- `[[S120:dc:isGiftAmountGreaterThanOrEqualTo:$50]]`
Returns the string value of the "isGiftAmountGreaterThanOrEqualTo" property of the object named by "dc", supplying "\$50" as an argument to the property method.
- `[[S120:directory:memberAttr:work_phone]]`
Displays the work phone of a constituent. Testing to see if the field 'work phone' contains a value can further customize this:
`[[?zz::z[[S120:directory:memberAttr:work_phone]]z:::
Work Phone
[[S120:directory:memberAttr:work_phone]]]]`

XML

```
<convio:session name="120" param=" dc:name"/>
```

S121 Map Property

Use the S121 tag to return the string representation for a named value of a map. The map must have been established using `MapValueSTagComponent.setMap()` beforehand.

For example, if you have a Map referenced by the variable `myMap` that contains `Boolean.TRUE` under the key "fooModeEnabled," and you call `MapValueSTagComponent.setMap(session, myMap)` before rendering, then `[[S121:fooModeEnabled]]` will return "true".

For a full list of the valid maps and tags available, the tag `[[S121:help]]` will render all valid maps and values for a given page.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
`[[S121: mapName:propertyName]]`
- **XML**
`<convio:session name="121" param=" mapName:propertyName" />`

Parameters

mapName - Optional. The name of the map from which to retrieve the property to render. A default map is used if this is not specified.

propertyName - The name of a property to render.

Sample Code

- **Standard Input**
`[[S121:api_map_param:Ship-Phone]]`
Conditionally render an error message.
`[[?xxnullxx::x[[S121:errorMsg]]x:::<p class="ErrorMessage">[[S121:errorMsg]]</p>]]`
- **XML**
`convio:session name="121" param="api_map_param:Ship-Phone" />`

S151 Determine Whether a Pagebuilder Page Exists

Use the S151 tag to test if a named page is available in the current security context.

Renders one of the string literals "TRUE" or "FALSE."

Note: S151 does not take the "category" or "wrap" options that S51 does since they are not relevant in calculating the response.

Application

PageBuilder

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S151:pagename]]
- **XML**
<convio:session name="151" param="pagename" />

Parameters

- *pagename* - The name of the page for which to test. That may be a PageNavigator-style path such as "topfolder/folder/pagename." Because the slash character was initially used as the separator for the category and is also the path separator for folders, the token after the final slash is interpreted as a category ID if it is numeric but the page name if it is not numeric.

Sample Code

- **Standard Input**
[[S151:homepage]]
- **XML**
<convio:session name="151" param="homepage" />

S188 Dynamic Ask Component Amount With Min. and Max. Values

Use the S188 tag to render a monetary sum returned by a Dynamic Ask Component. S188 supports minimum and maximum values, and copies Donations 2.0 behavior.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S188:constituent_field,default_amount,modify_percentage,increment_amount, round_to_nearest,minimum_amount,maximum_amount]]
- **XML:**
<convio:session name="188" param="largest_trans_amount,\$5.00, -85, \$0.00,\$1.00,\$10.00,\$20.00"></convio:session>

Parameters

- *starting_amount* - A monetary amount that can be pulled from the constituent's record via an S1 tag. Using the WYSIWYG, this presents transaction information such as "largest_gift_amount." If a site has a custom monetary field enabled you can use that instead, such as "custom_monetary_amount3."
- *default_amount* - A monetary amount that will act as a default if the value from constituent field is null or 0.
- *modify_percentage* - A percentage by which to modify the starting_amount. May be positive to increase or negative to decrease the amount.
- *increment_amount* - After applying the modify_percentage to starting_amount, increment the value by this amount.
- *round_to_nearest* - Round the amount to the nearest multiple of this value.
- *minimum_amount* - Minimum amount if constituent field has non-zero value.
- *maximum_amount* - Maximum amount if constituent field has non-zero value. If this value is not specified or specified value is 0, value 10 million is used.

Sample Code

[[S188:largest_trans_amount,\$5.00,-85,\$0.00,\$1.00,\$20.00,\$80.00]]

1. Take the constituent's largest previous transaction. If this value is 0 or null, return default value (\$5.00). For this example, we'll use a largest previous transaction of \$100.
2. Increment that amount by -85%: $100 + (100 * -0.85) \rightarrow 100 + (-85) \rightarrow 100 - 85 == \15.00
3. Add \$0.00: $15 + 0 = \$15.00$
4. Round to the nearest multiple of \$1.00 = \$15.00
5. Take maximum of \$15.00 and \$20.00 (specified minimum): $\max(\$15.00, \$20.00) = \$20.00$
6. Take minimum of \$20.00 and \$80.00 (specified maximum): $\min(\$20.00, \$80.00) = \$20.00$

[[S188:custom_monetary_value3,\$10.00,10,\$7.00,\$5.00,\$20.00,\$50.00]]

- Assume that the site has turned on custom_monetary_value3 and that it holds \$50.00 for this constituent
\$50.00
 $50 + (50 * 0.1) = 55$
 $55 + 7 = 62$
round to nearest multiple of 5 = \$60
 $\max(\$60, \$20) = \$60$
 $\min(\$60, \$50) = \$50$
- Assume that the next constituent has 0.00 for custom_monetary_value3
Use default value \$10

S201 Generate Help Topic Link

Use the S201 tag to generate a URL for help topic pages. Depending on the user's location setting, the S tag can link to help that is translated into the user's language.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S201:helpTopic:topicAnchor]]
- **XML**
<convio:session name="201" param="helpTopic:topicAnchor"/>

Parameters

- *helpTopic* - The name of the help topic to display by the resulting URL.
- *topicAnchor* - (optional) Names an HTML anchor in the help topic page.

Sample Code

- **Standard Input**
[[S201:Admin_WorkingWithContentTypes]]
- **XML**
<convio:session name="201" param="Admin_WorkingWithContentTypes" />

S202 GIGYA Log In Component

Use the S202 tag to render a link to log in with credentials for an external site using Gigya.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S202]]
- **XML**
<convio:session name="202" />

Parameters

None

Sample Code

- **Standard Input**
[[S202]]
- **XML**
<convio:session name="202"></convio:session>

S203 Social Sharing Component

Use the S203 tag to render a link to log in with credentials for an external site using Gigya.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S203:comments|title|description|linkBackUrl|actionLinkText|actionLinkUrl|imageSrc|imageHref|linkText|providerId1,providerId2,...|componentGroup]]
- **XML**
<convio:session name="203"
param="comments|title|description|linkBackUrl|actionLinkText|actionLinkUrl|imageSrc|imageHref|linkText|providerId1,providerId2,...|componentGroup"/>

Parameters

- *comments* - Default comments that the user can edit
- *title* - Share title
- *description* - Share description
- *linkBackUrl* - Link target for title
- *actionLinkText* - Text that appears in the Action section of Facebook
- *actionLinkUrl* - Target for the Action section of Facebook
- *imageSrc* - URL of the image shared on Facebook
- *imageHref* - Target of the image of Facebook
- *linkText* - Text that precedes the social network icons in the component (ie, Share this on [Icon][Icon][Icon])
- *providerId1,providerId2* - IDs of the social network providers to be used in the component. Currently supported IDs: email, Facebook, LinkedIn, Myspace, Twitter, Yahoo.
- *componentGroup* - An arbitrary name used to group this component with others for reporting purposes

For more information on how each of these fields displays on specific social networks see:

http://wiki.gigya.com/020_Developer_Guide/60_Publishing_User_Actions/010_The_Share_Widget/010_Advanced_Report

Sample Code

- **Standard Input**

```
[[S203:Comment goes here|Shasta Test Site|This is a really cool  
site.|CURRENT_PAGE_URL|Action Link|CURRENT_PAGE_URL|  
http://vmedia.rivals.com/IMAGES/Player/video/112809-  
TCUANDYDALTON_250.JPG|CURRENT_PAGE_URL|Share With Friends  
In|facebook,linkedin,myspace,twitter,yahoo,email|Sharing1]]
```

- **XML**

```
<p><convio:session name="203" param="Comment goes here|Shasta  
TestSite|This is a really cool site.|CURRENT_PAGE_URL|Action  
Link|CURRENT_PAGE_URL|http://vmedia.rivals.com/IMAGES/Player/video/1  
12809-TCUANDYDALTON_250.JPG|CURRENT_PAGE_URL|Share With  
Friends In|facebook,linkedin,myspace,twitter,yahoo,  
email|Sharing1"></convio:session></p>
```


S311 Renders a Navigation Bar Link

Use the S311 to get the URL from a specified navigation bar link. Embed the URL as a link or construct a customized navigation bar, and so forth.

Application

Navbar

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S311:LINK_ID]]
- **XML**
<convio:session name="311" param=" LINK_ID" />

Parameters

- *LINK_ID* - ID of a component (such as "Events") on a navigation bar. Sample Code

Standard Input

[[S311:LINK_ID]]

S330 TeamRaiser Check-In Status

Use the S330 tag to render a Teamraiser participant's check-in status. Possible output the tag returns:

- "Invalid" - no registration found
- "Initial" - initial check-in status
- "Started" - checkin started
- "Committed" - finished checkin, pledged but not paid
- "Paid" - finished check-in, paid
- "Complete" - finished check-in, no further action required
- "Unknown" - unknown state

Application

TeamRaiser

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S330:fr_ID]]
- **XML**
<convio:session name="330" param="fr_ID" />

Parameters

- *fr_ID* - Teamraiser event ID. If not provided, attempts to read fr_ID parameter from request.
- *cons_ID* - Constituent ID

Sample Code

Standard Input

```
[[?xcommittedxpaidxcompletex::x[[S330]]x::Check-in Complete!::Check-in  
Incomplete.]]
```

S334 Renders HTTP Query Data

Use the S334 tag to return the value of a parameter in the HTTP request.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**
[[S334:url_param_name]]
- **XML**
<convio:session name="334" param="url_param_name"/>

Parameters

url_param_name - The name of a parameter in the URL. Only parameters that exist in the URL are allowed. Post parameters will not be displayed.

Sample Code

If the homepage on support101.convio.com had an S334 tag for the parameter "foo" (written as [[S334:foo]]), then the output of the page would vary as follows, depending on the URL used to access the homepage.

URL	Output
http://support101.convio.com/site/PageServer	Empty string
http://support101.convio.com/site/PageServer?foo=bar	bar
http://support101.convio.com/site/PageServer?pg=homepage&foo=bar	bar
http://support101.convio.com/site/PageServer?foo=fish	fish

S335 Campaign Type

Use the S335 tag to distinguish between TeamRaiser and Tribute campaigns.

Application

TeamRaiser

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**
[[S335]]
- **XML**
<convio:session name="335"/>

Parameters

None

Sample Code

```
[[?xtributex::x[[S335]]x]]
```

[[S335]] attempts to read a valid TeamRaiser ID from the request parameters by first checking fr_ID and then checking FR_ID. If it finds a valid TeamRaiser ID, it loads up the TeamRaiser Campaign and prints campaign type. Often used to determine if a given TeamRaiser is a Tributes campaign and, if so, to render appropriate content.

Possible Output will be one of the following:

empty string if no TeamRaiser Campaign is found

"classic" if it is a classic TeamRaiser

"tribute" if it is a Tributes Campaign

Conditional

This conditional uses the S335 tag to determine whether the application is a Tribute and to resolve whether or not content should be displayed.

```
[[?SHOW::[[?xtributex::x[[S335]]x::[[?xTRUEx::x[[S100:TRIBUTES_CHAMPION_ALLOW_CARD_PROCESS]]x::SHOW::HIDE]]::[[?xTRUEx::x[[S400:0:F2F_PAR
```

TICIPANT_ALLOW_CARD_PROCESS]]x::SHOW::HIDE]]]]::<input type="radio"
name="off_pay_type" ID="off_pay_type_3" value="3" />:::]

S336 Banner Field

Use the S336 tag to display a banner field associated with an outreach campaign.

Application

Widgets

WYSIWYG Equivalent

Widgets > General

CMS Compatible

No

Format

- **Standard**
[[S336:outreach_campaign_ID:component_ID]]
- **XML**
<convio:session name="336" param="outreach_campaign_ID:component_ID"/>

Parameters

outreach_campaign_ID - The ID of the campaign with which the banner is associated.

component_ID - The ID of the specific widget.

Sample Code

- **Standard Input**
[[S336:1001:1002]]
- **XML**
<convio:session name="336" param="1001:1002"/>

S337 Progress Field

Use the S337 tag to render a progress field.

Application

Widgets

WYSIWYG Equivalent

Progress Meters > TeamRaiser Campaigns

CMS Compatible

Yes

Format

- **Standard**
[[S337:application type:objectId:subObjectId:consId:goalAmount:progress image ID:text color]]
- **XML**
<convio:session name="337" param="application type:objectId:subObjectId:consId:goalAmount:progress image ID:text color" />

Parameters

- *application type* - one of the following: teamraiser_campaign, donations_campaign, advocacy_alert
- *objectId* - fId, ID on advocacy alert table
- *subObjectId* - Team ID
- *consId* - The constituent ID
- *goalAmount* - number
- *progress image ID* - The ID of a widget image to use
- *text color* - The color of the text

Sample Code

- **Standard Input**
[[S337:teamraiser_campaign:1013:0:0:0:1:#000000]]
- **XML**
<convio:session name="337" param="teamraiser_campaign:1000:0:0:0:1:#000000"></convio:session>

S345 Top 10 List for TeamRaiser National Companies

Use the S345 tag to display a list of the top 10 performers among TeamRaiser National Companies.

Application

TeamRaiser

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S345:company_ID:list_type:limit]]
- **XML**
<convio:session name="345" param="company_ID:list_type:limit" />

Parameters

- *company_ID* - This is the ID of the national company to use. If none is provided, the S tag will fall back to parsing the company_ID parameter from the request.
- *list_type* - This determines which top 10 list you will be using. Valid choices are:
 - participants
 - teams
 - events
- *limit* - The maximum number of results to return.

Sample Code

- **Standard Input**
[[S345:1080:teams:5]]
- **XML**
<convio:session name="345" param="1080:teams:5"></convio:session>

S346 National Company Gift Summary

Use the S346 tag to display a National Company Gift Summary.

Application

TeamRaiser

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S346:company_ID]]
- **XML**
<convio:session name="346" param="company_ID" />

Parameters

company_ID - The unique ID of a national company. If none is provided, this STag will fall back to parsing the company_ID parameter from the request.

Sample Code

- **Standard Input**
[[S346:1080]]
- **XML**
<convio:session name="346" param="1080"></convio:session>

S347 TeamRaiser National Company Information

Use the S347 tag to render a component that cycles through the national companies for a specified TR type and renders the logo of each (with a fade effect as a transition). Each image is linked to the appropriate national company.

Application

TeamRaiser

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**
[[S347:tr_type:seconds]]
- **XML**
<convio:session name="347" param="tr_type:seconds" />

Parameters

- *tr_type* - The TeamRaiser type for which you wish to show sponsors. If none is provided, the tag will parse fr_ID from the request parameters and load the sponsors for the TeamRaiser Type of the event.
- *seconds* - The number of seconds to display each logo.

Sample Code

- **Standard Input**
[[S347:3:4]]
- **XML**
<convio:session name="347" param="3:4"></convio:session>

S348 TeamRaiser Dynamic Progress

Use the S348 tag to render number of participants, number of teams, and total dollars for a TeamRaiser event in a rotating, dynamic way.

Application

TeamRaiser

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S348: seconds]]
- **XML**
<convio:session name="348" param="seconds" />

Parameters

seconds - The number of seconds before switching to the next statistic.

Sample Code

- **Standard Input**
[[S348:10]]
- **XML**
<convio:session name="348" param="10"></convio:session>

S349 Total Dollars Raised for a TeamRaiser Event

Use the S349 tag to render total dollars raised for a TeamRaiser event in a dynamic way.

Application

TeamRaiser

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S349]]
- **XML**
<convio:session name="349" />

Parameters

None

Sample Code

- **Standard Input**
[[S349]]
- **XML**
<convio:session name="349"></convio:session>

S350 TeamRaiser Event Manager Permission

Use the S350 tag to render "True" or "False" to determine what (if any) permissions an event manager has for a TeamRaiser event.

Application

TeamRaiser

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S350:fr_ID:permission]]
- **XML**
<convio:session name="350" param=" fr_ID:permission" />

Parameters

- *fr_ID* - Campaign ID. If none is provided, the component parses from request parameters.
- *permission* - The object permission to check for. If none is provided, the component checks for ANY event manager permissions.

Sample Code

- **Standard Input**
[[S350]] checks for any event manager permission for the contextual event.
[[S350:1000]] checks for any event manager permission for event 1000.
[[S350:1000:FOO]] checks for the FOO permission for event 1000.
Note: [[S350]] alone will return true or false if the logged in user is an event manager. In the conditional below, different content is displayed depending on the user's status as administrator or event manager.
[[?xfalsex::x[[S350]]x::[[C89:admin_html_editor]]::[[C89:emgmt_html_editor]]]]
- **XML**
<convio:session name="350" param="1000"></convio:session>

S355 Google Map

Use the S355 tag to render Google Maps in web pages. This component allows the showing of one or many markers for single or multiple locations (only one Google map is supported in each page). It also provides a link for users to find directions to those locations. Map width, height, and zoom level are supported. Autozoom (finds the optimal zoom level based on the markers given) is provided if given a -1 zoom level. **Caution:** Do not use this tag on a secure site.

Google Maps requires several site options.

- *GMAP_ENABLED* = TRUE
- *GMAP_API_ADMIN_KEY* = API key for administrative side, supplied by Google at <http://code.google.com/apis/maps/signup.html>
- *GMAP_API_USER_KEY* = API key for the user side domain name if different from the administrative side, also obtainable at <http://code.google.com/apis/maps/signup.html>
- *GMAP_BASE_URL* = Offered in case a secure proxy is to be used; otherwise, not needed

Google Map API documentation is available at <http://code.google.com/apis/maps/documentation/index.html>

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**
[[S355:address*title*description*color*width*height*zoom*map controls]]
- **XML**
<convio:session name="355"
param="address*title*description*color*width*height*zoom*map controls" />

Parameters

Note: The tokens are separated with an asterisk (*), which are separated from each other by a semicolon (;). For multiple address maps, each address, title, and marker color is separated with a semicolon (see examples, below). Each multiple map control is also separated with a semicolon.

- *address* - must include at least one of state, zip, and country; city and street are optional with state and zip
- *title* - descriptive title to render in the map
- *description* - brief (because embedded in the map itself) text contextualizing the map
- *color* - color of the markers
- *width* - integer in pixels for display
- *height* - integer in pixels for display
- *zoom* - initial zoom level
- *controls* - semicolon-separated list of map controls

Sample Code

The S355 tag has been fully integrated into the WYSYWIG. Follow the steps below to construct an S355 tag automatically.

Note: This example uses TeamRaiser.

1. Configure the SDPs as explained above.
 2. Create a TeamRaiser.
 3. Edit that TeamRaiser,
 4. On the Customize Pages step, edit the greeting page.
 5. Scroll to the WYSYWIG and select Components -> Extensions -> Google Map
 6. The Embedded Component Selector for Google Map opens.
 7. Select width, height, initial zoom level, and desired additional controls.
 8. Enter a location in the Address input.
 9. Provide a title.
 10. Enter a description.
 11. Select the desired color for the map markers.
 12. Click the Add More Locations button if desired. Additional location forms appear.
 13. When you are finished, click the **Insert** button.
 14. The component will render in the WYSIWYG as a cream box.
 15. Click preview to verify it looks the way you expect.
- **Standard Input**
Single-address example, creates a map centered on Austin TX, with a blue marker, 300x300px map, auto zoom level, and a small map control:

```
[[S355:Zilker Park, Austin, TX*Zilker!*Austin's weirdest  
park!*purple*400*400*3*GSmallMapControl;GScaleControl]]
```

Multi-address example that creates a map with both Austin TX and Dallas TX:

```
[[S355:AustinTX;DallasTX*Austin;Dallas*KeepAustinWierd;ThinkBig*blue;red*  
300*300*-1*GSmallMapControl;GScaleControl;GMapTypeControl]]
```

- **XML**

```
<convio:session name="355" param="Zilker Park, Austin, TX*Zilker!*Austin's  
weirdest park!*purple*400*400*3*GSmallMapControl;GScaleControl">  
</convio:session>
```

S356 YouTube Video

Use the S356 tag to render a YouTube video for end users. (Should not be used on a secure site as content served up from YouTube is not secure.) Before adding videos to your site, please refer to the YouTube terms of use, <http://www.youtube.com/t/terms>.

Application

Platform

WYSIWYG Equivalent

Extensions > YouTube Video

CMS Compatible

Yes

Format

- **Standard**
[[S356:ID_EMBED*****]]
- **XML**
<convio:session name="356" param="ID_EMBED*****"/>

Parameters

- *YouTube Video URL* - link on youtube.com, will include "/v/" in the string
- *Height* - pixel height
- *Width* - pixel width
- *Is Auto Play* - should the video play automatically when the page loads? (TRUE or FALSE)
- *Is Show Related* - should related videos appear at the end of the video and on mouseover (TRUE or FALSE)
- *Window Mode* - defaults to opaque, but can also use window and transparent (this parameter is the wmode attribute of the embedded flash)

To get YouTube videos working properly on your site, you need set the site option YOUTUBE_ENABLED to TRUE. Once this is configured correctly, you can insert the S356 tag into a WYSIWIG.

Each parameter value is separated with an asterisk (*).

Sample Code

- **Standard Input**

```
[[S356:ID_EMBED*http://www.youtube.com/v/lsO6D1rwrKc*320*270*false*true*opaque]]
```

- **XML**

```
<convio:session name="356"  
param="ID_EMBED*http://www.youtube.com/v/lsO6D1rwrKc*320*270*false*true*opaque "></convio:session>
```

S364 YouTube Video Bar

Use the S364 tag to render a YouTube video bar. This is a list of videos that appear in bar form, either vertical or horizontal. (Should not be used on a secure site as content served up from YouTube is not secure.)

Application

Platform

WYSIWYG Equivalent

Extensions > YouTube Video Bar

CMS Compatible

Yes

Format

- **Standard**
[[S364:PARAMETERS]]
- **XML**
<convio:session name="364" param="PARAMETERS" />

Parameters

- *CHANNEL* - Specifies a defined YouTube channel
- *SEARCH_TERMS* - Search for keywords associated with videos
- *IS_VERTICAL* - True or False
- *PLAYER_HEIGHT* - Display height in pixels
- *PLAYER_WIDTH* - Display width in pixels
- *THUMBNAIL_SIZE* - SMALL or MEDIUM

Sample Code

- **Standard Input**
[[S364:CHANNEL;GoogleDevelopers|SEARCH_TERMS;calendar|IS_VERTICAL;TRUE|PLAYER_HEIGHT;260|PLAYER_WIDTH;320|THUMBNAIL_SIZE;SMALL]]
- **XML**
<convio:session name="366" param="CHANNEL;GoogleDevelopers|SEARCH_TERMS;calendar|IS_VERTICAL;TRUE|PLAYER_HEIGHT;260|PLAYER_WIDTH;320|THUMBNAIL_SIZE;SMALL" />

S366 Facebook Connect Login Component

Use the S366 tag to render a login button for Facebook Connect. The button opens a popup window that prompts the user to log in.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**
[[S366:BUTTON_IMAGE_TYPE]]
- **XML**
<convio:session name="366" param="BUTTON_IMAGE_TYPE" />

Parameters

BUTTON_IMAGE_TYPE

Either of:

- *LIGHT_MEDIUM_SHORT* - to render a button with the Facebook "f" logo and the word "Connect."
- *LIGHT_MEDIUM_LONG* - to render a button with the Facebook "f" logo and the phrase "Connect with Facebook."

Sample Code

- **Standard Input**
[[S366:BUTTON_IMG_TYPE;LIGHT_MEDIUM_SHORT]]
- **XML**
<convio:session name="366"
param="BUTTON_IMG_TYPE;LIGHT_MEDIUM_SHORT" />

S400 Renders the Value of a TeamRaiser Data Parameter

Use the S400 tag to render content based on a customer's specific TeamRaiser event.

Application

TeamRaiser

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**
[[S400:fr_ID:tag:default_value]]
- **XML**
<convio:session name="400" param=" fr_ID:tag:default_value "/>

Parameters

- *fr_ID* - The TeamRaiser ID. If the value is 0, the S tag will try to get it from the request.
- *tag* - The name of the EDP, for example, F2F_ACCEPT_OFFLINE.
- *default_value* - Default value of the EDP (optional).

Note: EDP is a TeamRaiser Event Data Parameter defined for a specific TeamRaiser and overriding the site options.

Sample Code

Given a custom EDP that points to the location of an image, event managers or division managers can use that custom EDP to reference the location of a sponsor logo for their specific event. This allows each event to have its own sponsor logo, which can easily be changed out.

```

```


S402: TEAMRAISER HORIZONTAL NAVIGATION BAR

Use the S402 tag to render the TeamRaiser navigation section as a horizontal row of dropdown menus.

Application

TeamRaiser

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**
[[S403:event type:fiscal year]]
- **XML**
<convio:session name="403" param="event type:fiscal year"/>

Parameters

- *fr_ID* - the ID of a TeamRaiser other than the current one
- *0 (zero)* - use the ID of the current TeamRaiser

Sample Code

- **Standard Input**
[[S402:0]]
- **XML**
<convio:session name="402" param="0"></convio:session>

S403 TeamRaiser Program Type Tally

Use the S403 tag to render a tally encompassing the total amount and gift count.

Application

TeamRaiser

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**
[[S402:fr_ID]]
- **XML**
<convio:session name="402" param=" fr_ID " />

Parameters

- *event type* - the event type must be specified, this is the event/classification type
- *fiscal year* - the fiscal year of teamraisers to include in the tally totals

Sample Code

A TeamRaiser may be assigned an Event Type. This event type appears in the first screen when editing the TeamRaiser. The event type will be either a user-defined value or a value selected from the dropdown box. A fiscal year may also be assigned to the event.

Given two event types defined, *event_type1* and *event_type2*, and two fiscal years assigned to TeamRaisers *2008* and *2009*:

- **Standard Input**
[[S403:event_type1]]
gives a tally of all of the events that are assigned event_type1 and are active regardless of what fiscal_year is assigned (or not assigned) to the event.
[[S403:event_type1:2008]]
gives the same group of events as the previous example but includes only events assigned to that fiscal_year.
- **XML**
<convio:session name="403" param="event_type1"></convio:session>
<convio:session name="403" param=" event_type1:2008"></convio:session>

S404 Donor Wall Displaying the 5 Most Recent Team or Event Donors

Use the S404 tag to render a donor wall listing the five most recent donors to an team or an event. If the wall is for a team, the display includes gifts made directly to that team and to team participants. If for an event, the list includes all teams and gifts relating directly to that event. A link at the bottom of the control opens a new window that can display a searchable list of all donors.

Application

TeamRaiser

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S404:type:ID:title]]
- **XML**
<convio:session name="404" param="type:team,ID:1011,title:Recent Team Gifts"/>

Parameters

type - either team or event, required

event ID - the ID of the event

team ID - the ID of the team

title - title that will render at the bottom of the control

Sample Code

- **Standard Input**
[[S404:team,1011,1000,Recent Team Gifts]]
- **XML**
<convio:session name="404" param="1011,1000,Recent Team Gifts"></convio:session>

S405 Renders a Milestone Value

Use the S405 tag to render the value of a milestone, from a Milestone ID and a parameter name.

Note: the tag and its parameters are case-sensitive, all parameters are required, and MultiLocale uses are supported.

Application

TeamRaiser

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S405:milestone_ID,param_name]]
- **XML**
<convio:session name="405" param="milestone_ID,
param_name"/>

Parameters

- *milestone ID* - the ID of the milestone to render
- *param name* – Can be one of the following:
 - *amount*
 - *adminNotificationContent*
 - *adminNotificationFrequency*
 - *adminNotificationRecipients*
 - *adminNotificationRecipientsEM*
 - *description*
 - *mId* - (milestone ID)
 - *name*
 - *participantNotificationContent*
 - *participantNotificationContentText*
 - *participantNotificationEnabled*

- *participantNotificationSubject*

Sample Code

- **Standard Input**
[[S405:1000,amount]]
- **XML**
<convio:session name="405" param="1000,adminNotificationContent"/>

S406 Show Report Writer Results

Use the S406 tag to render a report's result table.

Note: this tag does not run the report.

Application

Reports

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**
[[S406:result_id]]
- **XML**
<convio:session name="406" param="result_id"/>

Parameters

result_id - (Required) Report results used for rendering the table.

Sample Code

- **Standard Input**
[[S406:1000]]
- **XML**
<convio:session name="406" param="1000"/>

S407 Show Amount Raised by TeamRaiser Donations

Use the S407 tag to render the amount of money that an event has raised from sponsors, confirmed donations, and unconfirmed donations. The unconfirmed donations are an optional parameter.

Application

TeamRaiser

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S407:FR_ID{optional:includeUnconfirmed}]]
- **XML**
<convio:session name="407" param="{optional:includeUnconfirmed}"/>

Parameters

- *FR_ID* - The name of the help topic to display by the resulting URL.
- *includeUnconfirmed* - Names an HTML anchor in the help topic page.

Sample Code

- **Standard Input**
[[S407:1001]]
[[S407:1001:false]]
[[S407:1001:true]]
- **XML**
<convio:session name="407" param="1001"/>
<convio:session name="407" param="1001:false"/>
<convio:session name="407" param="1001:true"/>

S408 Search Form (with Results) for TeamRaisers

Use the S408 tag to render a search form, with results, for TeamRaisers. Search criteria include event name, city, state, or distance by zip (USA only). When the request comes in, if there is search data in the request, the results of the search are displayed beneath the form. Because of this, you should not use this S tag on a page that cannot be refreshed (such as a Thank you page for a transaction).

Note: More than one S408 tag (search form) can be embedded in the same page. When one form is submitted, all instances of the S408 tag display the same results after the page refreshes.

Application

TeamRaiser

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard:**

```
[[S408:include_name:include_city:include_county:include_state:include_postal:  
comma_separated_list_of_radius_values:units_of_distance:category:classification:  
results_per_page]]
```

- **XML:**

```
<convio:session name="408"  
param="name:include_city:include_county:include_state:include_postal:comma_  
separated_list_of_radius_values:units_of_distance:category:classification:  
results_per_page"></convio:session>
```

Parameters

- *include_name* - A value of "0" means the form will not render the field allowing users to search by Event Name. Any other value will result in the event name field being available. It is recommended that you use either "0" or "name" for this parameter.
- *include_city* - just like name, but for the field that searches by Event City. A 0 means the field will not be displayed. Recommended: "0" or "city".
- *include_county* - just like name and city, but for the field that searches by Event County. A 0 means the field will not be displayed. Recommended: "0" or "county".

- *include_state* - like name and city, but for the state search field. Recommended: "0" or "stateprov".
- *include_postal* - like name, city, and state, but for the zip code field and distance search. Recommended: "0" or "postal"
- *comma_separated_list_of_radius_values* - This is a comma separated list of numbers that will be presented to the user in the selection form for distance searches. Valid numbers are 0-200, inclusive. Values under 0 or over 200 will be ignored when the component renders. If no valid values are provided, will attempt to use the values in the SDP F2F_EVENT_SEARCH_RADIUS_VALUES. If the SDP contains no valid numbers, the component will use a default value of "5,10,15,25,50,100,200". Duplicate entries in the list will not be displayed as duplicate choices to the site visitor. Entries will appear in the order that they are entered (they are not automatically sorted - ie a list of 10,5,20 will display to the user as a list of 10,5,20).
- *units_of_distance* - Will limit the distance search to this unit of measurement, or will allow the user to select whether they want to search by miles or kilometers. Accepted values are: mi, km, or choice. If an invalid option is passed in, the component will use the value of the SDP F2F_EVENT_SEARCH_DISTANCE_UNITS. If the SDP does not contain one of the 3 valid options, the system will default to using choice.
- *category* - Limits the search results to events within this category, a value of 0 will search all security categories. Accepts either the category_ID or the category Name. If there is no category with the specified category_ID or name, the component will render an error message. May be used in conjunction with the classification parameter to limit the search to events within a specific category and within a specific event type.
- *classification* - Limits the search results to events with a Primary Event Type (also known as a classification) equal to the specified parameter. A value of 0 will search all event types. Accepts either the classification ID or the classification name. If there is no Primary Event Type with the specified classification ID or name, the component will render an error message. May be used in conjunction with the category parameter to limit the search to events within a specific category and within a specific event type.
- *results_per_page* – The number of rows will to display per page in the search results. Valid ranges are 1-50, inclusive. If an invalid value is passed in, the component will attempt to use the value from the SDP F2F_EVENT_SEARCH_RESULTS_PER_PAGE. If the SDP contains an invalid value, the component will default to a value of 20 results per page. You cannot display more than 50 results per page. Values under 5 are not recommended as they provide bad usability, but values under 5 are great for testing.

- *include_public_event_type* - This field is optional. A value of "0" or "" means the form will not render the field allowing users to search by Public Event Type. Any other value will result in the Public Event Type search criteria field being available, provided that the related SDP (F2F_PUBLIC_EVENT_TYPES_ENABLED) is also enabled.. It is recommended that you use either "0" or "public_event_type" for this parameter. (Supported as of Aspen.)

Sample Code

- **Standard Input:**

```
[[S408:name:city:county:state:zip:5,10,15,20,50,100:choice:0:0:20]]
```

Renders a form that presents all search fields, searches all categories and event types, and allows the user to choose whether to search by miles or kilometers. Values that will appear in the distance search selection are: 5,10,15,20,50,100. Search Results will display 20 records per page.

```
[[S408:name:0:0:0:Postal:10,20,30,40,50:mi:General:Walks:5]]
```

Renders a form that only allows the user to search by name or postal code (including distance), limited to events in the "General" security category that are of the Primary Event Type "Walks." Limits the distance search to miles, with possible values in the distance selection of 10,20,30,40,50. Search results will display 5 records per page.

- **XML:**

```
<convio:session name="408"
param="name:city:county:state:zip:5,10,15,20,50,100:choice:0:0:20" />
```

The code may generate any of several error codes, as below:

bad_category_ID

There is no Security Category matching the ID provided.

bad_event_type

There is no Primary Event Type matching the classification ID provided.

not_enough_params

There are not enough parameters provided to correctly use this component. All parameters are required.

no_search_fields

The form has been configured to hide all available search fields from the user, resulting in no way for the user to actually search. Please enable at least 1 search form field.

S409 Renders an Instance of Participant Center 2 in a Convio Custom Page

Use the S409 tag to embed a Participant Center 2 page in a Convio custom page.

Application

TeamRaiser

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**
[[S409]]
- **XML**
<convio:session name="409" />

Parameters

None

Sample Code

- **Standard Input**
[[S409]]
- **XML**
<convio:session name="409" param=""/>

S413 Provides a Link to the eCommerce Store and Product Associated with the Event

Use the S413 tag to create a link to an eCommerce store associated with an event.

Application

TeamRaiser

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**
[[S413:fr_id, link_text, proxy_type, proxy_id, store_id, product_id]]
- **XML**
<convio:session name="413" param="fr_id, link_text, proxy_type, proxy_id, store_id, product_id " />

Parameters

- **fr_id** - event_id, use 0 to pull from session
- **link_text** - Supply text to make a clickable link or leave empty for the plain URL
- **proxy_type** - Give the type name (participant / event / team) or the ID (20 / 21 / 22)
- **proxy_id** - The cons_id for participant, the team_id for team or event_id for event
- **store_id** - The id of the store or use 0 for the local EDP value
- **product_id** - The id of the product or use 0 for the local EDP value, leave empty to build a link directly to the store

Sample Code

- **Standard Input**

```
[[S413:1000, Click me, 20, 1001001, 1002, 1001]]
```

[[S413:1000, , 20, 1001001, 1002, 1001]] returns a plain URL

[[S413:1000, Go to the store, 20, 1001001, 1002]] links directly to a store

- **XML**

```
<convio:session name="413" param="1000, Click me, 20, 1001001, 1002, 1001"/>
```

```
<convio:session name="413" param="1000, , 20, 1001001, 1002, 1001"/>
```

```
<convio:session name="413" param="1000, Go to the store, 20, 1001001, 1002"/>
```

S414 Renders List of Affiliate Events

Use the S414 tag to render a list component with the event name, link to the event, number of teams and amount raised (respecting local event options).

Application

TeamRaiser

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**
[[S414]]
- **XML**
<convio:session name="414" />

Parameters

- **fr_id** - teamraiser event id
- **page size** - number to display in the list per page
- **renderEmptyList** - true (or blank) to indicate to render the list event if there are not affiliate events. false to render nothing when there are not affiliate events.

Sample Code

- **Standard Input**
[[S414:fr_id,count,renderEmptyList]] / [[S414:1000,10,FALSE]]
- **XML**
<convio:session name="414" param="1000,10,FALSE"/>

S415 Renders List of Affiliate Events

Use the S415 tag to render top affiliated teams, given a lead event id, in tabular format.

Application

TeamRaiser

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**
[[S415]]
- **XML**
<convio:session name="415" />

Parameters

- **lead_event_id** - the fr_id of the lead event to which to pull affiliated events from and teams from those affiliated events
- **include_lead_event** - (true/false) - whether or not to include the lead event's teams in the top affiliated teams list
- **num_teams** - the number of teams to include in the list
- **display_mode** - whether the STag should display donation amounts (SUM) or the number of donations (COUNT)

Sample Code

- **Standard Input**
[[S415:lead_event_id:include_lead_event:num_teams:display_mode]]
- **XML**
<convio:session name="415"
param="lead_event_id:include_lead_event:num_teams:display_mode" />

S421 Personal Events Search Field

Use the S421 tag to render an event search component. Criteria include event name, location, radius from the location, activity type, and date range.

Application

Personal Events

WYSIWYG Equivalent

Personal Event > Personal Events Search

CMS Compatible

Yes

Format

- **Standard**

```
[[S421:name:city:state:zip:1000:dateRange:5,10,15,25,50,100,200:1000]]
```

- **XML**

```
<convio:session name="421"  
param="name:city:state:zip:1000:dateRange:5,10,15,25,50,100,200:1000"></  
convio:session>
```

Parameters

- *name* - the name of the Personal Event, a string such as "main_event"
- *city* - city where the event will be held
- *state* - state in which the event will be held
- *zip* - zip code of the event location
- *activity type* - a system-generated code number for the activity
- *date range* - time period within which to search for events
- *radius* - maximum distance in miles from location provided in search

Sample Code

- **Standard Input**

```
[[S421:name:city:state:zip:1000:dateRange:5,10,15,25,50,100,200:1000]]
```

- **XML**

```
<convio:session name="421"  
param="name:city:state:zip:1000:dateRange:5,10,15,25,50,100,200:1000"></  
convio:session>
```


S422 Personalized Link in Emails From Event Host

Use the S422 tag to embed in emails a link back to the host's event page.

Application

Personal Events

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**

[[S422:page_tag:activity_ID:event_ID:sender_cons_ID:taf_ID:track_enabled]]

- **XML**

<convio:session name="422" param="event type:fiscal year"/>

Parameters

- *page_tag* - a system name for the page, such as "event_list"
- *activity_ID* - the ID of the activity
- *event_ID* - the ID of the Event
- *sender_cons_ID* - the cons_ID of the Host
- *taf_ID* - the type of message being sent
- *track_enabled* - a boolean value, 0 or 1, to indicate whether any click-through is to be tracked

Sample Code

- **Standard Input**

[[S422:0:2002:1003:3043:4001:1]]

- **XML**

<convio:session name="422"
param="0:2002:1003:3043:4001:1"></convio:session>

S500 Value of a Center-Specific Data Parameter (CDP)

Use the S500 tag to render the value of a named CDP. If a TeamRaiser Event Data Parameter (EDP) exists for this value, then that EDP will be evaluated first. If no override exists, then the CDP will be evaluated.

Application

Platform

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**

[[S500:center_ID:tag]]

- **XML**

<convio:session name="500" param="center_ID:tag" />

Parameters

- *center_ID* - The Center Security Category ID. If the value is 0, the S tag will try to get it from the request on's ID of the champion. If not supplied (or zero), the session user is used.
- *tag* - The name of the CDP, such as SITE_NAME.

Sample Code

- **Standard Input**

[[S500:1000:SITE_NAME]]

Renders the site name for the center with the ID of 1000

- **XML**

<convio:session name="500" param="1000:SITE_NAME"></convio:session>

S530 Champion Fund List

Use the S530 tag to create a list of champion funds. If the current user is a champion, the links go to the champion home page for that fund. If the current user is not a champion, the links go to the pages for the current user and no funds are listed.

Application

Personal Fundraising

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**

[[S530:champ_cons_id:campaign_id:display_option]]

- **XML**

```
<convio:session name="530"  
param="champConsId:campaignId:display_option"/>
```

Parameters

- *champ_cons_id* - The champions constituent ID. If there is none, then the session user is used.
- *campaign_id* - The campaign ID (fr_id) used to limit the results. If not supplied (or zero), then all enabled funds across all campaigns are shown for the champion.
- *display_option* - Optional parameter used to influence the layout of the results.
 - *list-by-campaign (default option)* - Renders a set of lists. The current Campaign will be listed first, followed by other campaigns. Each campaign name will be followed by a list of funds in that campaign.
 - *list* - Renders a standard HTML list
 - *list-include-campaign* - Renders a standard HTML list. The fund name is preceded by the campaign name plus a colon.

Sample Code

User 123 has fund A-100 and B-100 in campaign 100 and fund C-200 in campaign 200.

- If 123 is logged in, the links lead to the champion center.
- If 123 is not logged in, the links lead to the fund page.

Note: Decide if *current campaign* should be rendered instead of *campaign name* when rendering for a campaign.

[[S530]]

Renders all funds for the logged in user
Renders all of 123's funds ONLY if 123 is logged in

[[S530:123]]

Renders all of 123's funds.

[[S530:123:200]]

Renders only C-200

[[S530:200]]

Renders all campaign 200 funds for the logged in user
Renders C200 ONLY if 123 is logged in

[[S530:123::list-include-campaign]]

Renders all of 123's funds with a campaign name prefix:

- * 100: A-100
- * 100: B-100
- * 200: C-200

[[S530:123::list-by-campaign]]

Renders all of 123's funds by campaign:

- *100 Funds*
- * A-100
- * B-100
- *200 Funds*
- * C-200

[[S530:123::list-by-campaign]]

Renders all of 123's funds by campaign when 200 is the current campaign:

- *200 Funds*
- * C-200

```

*100 Funds*
* A-100
* B-100

[[S530:123:100:list-by-campaign]]

*100 Funds*
* A-100
* B-100

[[S530:123:200:list-by-campaign]]

Where 200 is not the current campaign

*200 Funds*
* C-200

```

CSS and Template Customization

CSS styles are used as follows for each display type:

list (default option) - Renders as standard HTML list using and with within a div. The div uses a space-separated list of CSS classes:

```
pf-champ-fund-list pf-champ-fund-list-display-list
```

list-include-campaign - Renders as standard HTML list using and with within a div. The div will use a space-separated list of CSS classes:

```
pf-champ-fund-list pf-champ-fund-list-display-list-include-campaign
```

list-by-campaign - Renders as a set of lists. Each campaign name is a standard HTML <h3>. It also has a standard HTML list using and enclosed within a div (<div>). Each div uses a space-separated list of CSS classes:

```
pf-champ-fund-list pf-champ-fund-list-display-list-by-campaign
```

Template customization is possible by updating the following:

row template

```
friendraiser/tributes/component/champion_fund_list_component_row.tpt
```

campaign/list container template

```
friendraiser/tributes/component/champion_fund_list_component.tpt.
```

S535 Personal Fundraising Campaign Status Thermometer

Use the S535 tag to render Personal Fundraising campaign status or fund status using a thermometer display.

Application

Personal Fundraising

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**

[[S535: type,frID,fundID]]

- **XML**

<convio:session name="535" param="type,frID,fundID "/>

Parameters

- *type*

- campaign_therm
- fund_therm
- team_therm

- *frID* - ID of the Personal Fundraising campaign

- *fundID* - ID of a fund in the Personal Fundraising campaign

Sample Code

- **Standard Input**

[[S535:team_therm, 1011,100]]

- **XML**

<convio:session name="530" param="team_therm,1011,100">
</convio:session>

S542 Personal Campaign Information

Use the S542 tag to render summary information for a Personal Fundraising campaign.

Application

Personal Fundraising

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**

[[S542: fr_ID:type]]

- **XML**

<convio:session name="542" param="fr_ID:type"/>

Parameters

fr_ID - the ID of the Personal Fundraising campaign

type

- *num-champions* - The number of champions registered for the campaign.
- *num-participants* (legacy) - The number of champions registered for the campaign. (use num-champions instead)
- *num-funds* - The number of funds created for the campaign.
- *num-teams* (legacy) - The number of funds created for the campaign. (use num-funds instead).
- *num-donors* - The number of people that have donated to any fund in the campaign.
- *num-donations* - The number of donations that have been made to the any fund in the campaign.
- *dollars* - The total amount of donations that have been made to the any fund in the campaign.
- *fr-ID* - the fr_ID for the campaign
- *donation-app-ID* - the donation app ID

- *campaign-ID* - the ID of the donation campaign
- *form-ID* - the ID of the donation form
- *title* - the campaign title
- *goal* - the campaign goal
- *to-reach-goal* - goal - dollars
- *tax-ID* - tax ID
- *status* - numeric value for status of campaign
- *type* (legacy) - Returns campaign type, which is always tribute.

Sample Code

- **Standard Input**

```
[[S542:1004:num-champions]]
```

- **XML**

```
<convio:session name="542" param="1004:num-champions"></convio:session>
```


S548 Fund-Specific Champion Information for a Personal Fundraiser

Use the S548 tag to render summary information for a Champion (specific to one fund).

Application

Personal Fundraising

WYSIWYG Equivalent

None

CMS Compatible

No

Format

- **Standard**
[[S548:fr_ID:type]]
- **XML**
<convio:session name="548" param="fr_ID:type"/>

Parameters

fr_ID - the ID of the Personal Fundraising campaign

type

- *part-type* (legacy support) - renders "Champion" if the session user is champion for the current fund (use if-is-champion instead)
- *fund-ID* - The ID of the fund
- *team-ID* (legacy support) - The ID of the team or blank if not on a team (use fund-ID instead)
- *fund-name* (legacy support) - The name of the fund
- *team-name* (legacy support) - The name of the fund (use fund-name instead)
- *if-is-champion* - If the session user is champion of the current fund, TRUE
- *if-is-captain* (legacy support) - If the session user is champion of the current fund, TRUE. (use if-is-champion instead)
- *if-on-team* (legacy support) - If the session user is champion of the current fund, TRUE. (use if-is-champion instead)
- *fr-ID* - Renders the fr_ID for the fund's campaign

- *cons.FIELD* - Renders the given field for the champion's constituent record

Sample Code

- **Standard Input**
[[S548:1003:fr-ID]]
- **XML**
<convio:session name="548" param="1003:fr-ID"></convio:session>

S549 True or False if the User is a Champion of Any Fund

Use the S549 tag to determine whether the user is a fund champion.

Application

Personal Fundraising

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S549:conslid]]
- **XML**
<convio:session name="549" param="conslid"/>

Parameters

conslid (optional) - The consld of the user to check. If blank or zero references the session user.

Sample Code

```
[[?TRUE::[[S549]]::  
<strong>Thank you for being a Champion</strong>  
<br/>
```

You can choose a fund to manage from your list below, or by using the Champion Dashboard link

```
<br/>
```

```
[[S530]]
```

```
::
```

You can join [[S542:0:num-donors]] other donors by clicking on Support a Cause

```
<br/ >
```

```
<strong>Want to help even more?</strong>
```

```
<br/>
```

You can join [[S542:0:num-champions]] other Champions by clicking on Champion a Cause]]

S550 HTML Option Tags with Years to Select

Use the S550 tag to configure the year range in a dropdown menu.

Application

UI Controls

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S550:BD=[Backwards Differential]:FD=[Forward Differential]:SV=[Selected Value]:FV=[Future View]]]
- **XML**
<convio:session name="" param="BD=[Backwards Differential]:FD=[Forward Differential]:SV=[Selected Value]:FV=[Future View]"/>

Parameters

BD (Backward Differential) - The value for this parameter is an integer indicating how many years before the current year that the drop-down menu should display. If the current year is 2010 and a value of -10 is provided the drop-down menu will start at 2000. This parameter is optional. Default value for this parameter is 0.

FD (Forward Differential) - The value for this parameter is an integer indicating how many years beyond the current year that the drop-down menu should display. If the current year is 2010 and a value of 10 is provided the drop-down menu will end at 2020. This parameter is optional. Default value for this parameter is 10.

SV (Selected Value) - The value for this parameter is an integer indicating which year should be selected by default. This parameter is optional. If this parameter is not specified, the first option is selected.

FV (Future Value) - The value for this parameter is a string value which can be either "true" or "false." This parameter represents whether the drop-down menu should display in descending or ascending order. If "true," the years in the drop-down menu will display in descending order, and if "false" the years will display in ascending order. The default value for this parameter is "false." This is an optional parameter.

Sample Code

For the following example, the option to be selected is stored in template parameter `[[A1]]` and the current year is 2010.

Default Usage:

```
[[S550 : : : SV = [[A1]]]]
```

This will result in a drop-down menu that runs from 2010 to 2019.

Future View Usage:

```
[[S550 : : : SV = [[A1]]:FV=True]]
```

This will result in a drop-down menu that runs from 2019 to 2010.

Specific Year Range:

```
[[S550 : BD = -10 : FD = 10 : SV = [[A1]]]]
```

This will result in a drop-down menu that runs from 2010 to 2019.

S551 Absolute URL

Use this tag to create an absolute URL from a relative URL. This tag is especially useful if you are creating content that will be displayed outside your site.

Application

Reports

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**

[[S551:relativeUrl]]

- **XML**

<Convio:session name="551" param="relativeUrl"/>

Parameters

relativeUrl – A relative URL from the ECRM page containing the S551 tag, to another resource in the ECRM product.

Sample Code

Standard Input

The two code samples below will link to a Captcha image in a Convio custom page or component. However, if you are designing HTML content that others will cut and paste into another file, only the second tag will work because it is rendered using an absolute URL.

```
<a href="Captcha">Link to Captcha</a>
```

```
<a href="[[S551:Captcha]]">Link to Captcha</a>
```

S565 Include Email in List

Use this tag to render a list of email messages by subject name. Clicking on a subject name allows users to view the email message content. The email messages included in this list are active messages that have been approved, sent, and flagged for inclusion in this tag. The list is sorted in descending order based on the delivery date. Under email campaign, in the list of email messages is an action link to include a message in a list, which invokes this tag.

Application

Email

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S565]]
- **XML**
<Convio:session name="551" param=""/>

Parameters

None.

Sample Code

Standard Input

Not applicable.

S1151 List of Letter to the Editor Alerts

Use the S549 tag to determine whether the user is a fund champion.

Application

Advocacy

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S1151:parameters]]
- **XML**
<convio:session name="1151" param="parameters"/>

Parameters

numLinks

Specifies the maximum number of alerts to display. Default: 999999.

issue

In the case where the list should only include alerts that are associated with a particular advocacy issue, specifies the ID of the folder corresponding to an advocacy issue. "-1" indicates there is no constraint. Default: -1.

desc

Specifies whether to display the alert description. Default: description. The potential values are:

- *Description* - Display the alert description.
- *no* - Don't display the alert description.

taken

Specifies whether to display alerts that have already been taken by the constituent. Default: *show*. The potential values are:

- *show* - Show alerts that have been taken.
- *hide* - Don't show alerts that have been taken.

link

Specifies whether to display the link separately or as part of the alert title. Default: *title*. The potential values are:

- *Title* - Make the alert title the link.
- *separate* - Display the link separately.

sortBy

Specifies how to sort the list. Default: *priority*. The potential values are:

- *priority* - Sort by alert priority.
- *expire* - Sort by alert expiration date.
- *publish* - Sort by alert publish date.

listStyle

Specifies the style in which the list should be rendered. Default: *plain*. The potential values are:

- *plain* - Display the list without any bullets or numbers.
- *bullet* - Display list as a bullet list.
- *number* - Display list as a numbered list.

Sample Code

```
[[S1151:numLinks:999999:issue:-  
1:taken:show:desc:description:sortBy:priority:link:title:listStyle:plain]]
```

S1152 LTE Alerts Taken by the Current User

Use the S1152 tag to render a list of LTE alerts taken by the current user. If the user has not participated in any LTE alerts, or if the user can not be determined (not logged in), nothing is displayed.

Application

Advocacy

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S1152:parameters]]
- **XML**
<convio:session name="1152" param="parameters"/>

Parameters

numlinks

Specifies the maximum number of alerts to display. Default: 999999

issue

In the case where the list should only include alerts that are associated with a particular advocacy issue, specifies the ID of the folder corresponding to the advocacy issue. "-1" indicates there is no constraint (Default: -1).

desc

Specifies whether to display the alert description. Default: *description*. The potential values are:

- *description* - Display the alert description.
- *no* - Don't display the alert description.

date

Specifies whether to display the date when the constituent took action. Default: *no*. The potential values are:

- *actiondate* - Display the date the constituent took action.
- *no* - Don't display the date the constituent took action.

stats

Specifies whether to display statistics for the alert. Default: *no*. The potential values are:

- *statistics* - Display statistics for the alert.
- *no* - Don't display statistics for the alert.

tafLink

Specifies whether to display a Tell-a-Friend link. Default: *title*. The potential values are:

- *none* - Don't display a Tell-a-Friend link.
- *title* - Make the alert title a Tell-a-Friend link.
- *separate* - Display the Tell-a-Friend link separately.

sortBy

Specifies how to sort the list. Default: *priority*. The potential values are:

- *priority* - Sort by alert priority.
- *expire* - Sort by alert expiration date.
- *publish* - Sort by alert publish date.
- *action* - Sort by the date the constituent took action.

listStyle

Specifies the style in which the list should be rendered. Default: *plain*. The potential values are:

- *plain* - Display the list without any bullets or numbers.
- *bullet* - Display list as a bullet list.
- *number* - Display list as a numbered list.

Sample Code

```
[[S1152:numLinks:10:issue:-  
1:desc:description:date:actiondate:stats:statistics:tafLink:title:sortBy:action:listStyle:plain]]
```

- *numLinks:10* - Displays a maximum of 10 links.
- *issue:1* - Displays all alerts whether or not they are associated with a particular advocacy issue.
- *desc* - description displays the description of each alert
- *date:actiondate* - Displays the date the constituent took action.

- *stats: statistics* - Displays statistics for each alert
- *tafLink:title* - Converts each alert title into a Tell-a-Friend link
- *sortBy:action* - Sorts the alert list by the date the constituent took action
- *listStyle:plain* - Displays the list without bullets or numbers

S1153 Lists Recipients of a Letter to the Editor Alert

Use the S1153 tag to render a list of the recipients of a LTE message sent by a (possibly) logged in constituent.

Application

Advocacy

WYSIWYG Equivalent

None

CMS Compatible

Yes

Format

- **Standard**
[[S1153]]
- **XML**
<convio:session name="1153"/>

Parameters

The parameters are a series of name-value pairs parsed based on their position in the list. The name-value pairs are separated by colons.

- *lte_ID* - LTE Alert to use (if not provided, the LTE Alert is determined from context)
- *response_ID* - Which response to use (if not provided, the constituent is determined from context and all responses are returned -- currently, LTE limits to one response per constituent)
- *recipients* - Which recipients to display:
 - *all* - display all recipients
 - *email* - display only email recipients (NOT IMPLEMENTED)
 - *fax* - display only fax recipients (NOT IMPLEMENTED)

Sample Code

- **Standard Input**
[[S1153:lte_ID:0:response_ID:0:recipients:all]]
- **XML**
<convio:session name="1153"
param="lte_ID:0:response_ID:0:recipients:all"/>

Appendix A: S120 Object Properties

The S120 tag returns the string value of a named property from a named object based on the Convio application. For example, the S120 displays the values for the following Autoresponders and Thank You pages.

Donation2 (General Donations) Autoresponders

The S120 tag supports two categories of donation autoresponder.

Contact Information

- Name: [Donor's name]
- Address: [Donor's address]
- Email Address: [Donor's email address]

Donation Information

- Donation Form: [Donation form name]
- Donation Level: [Donation level name]
- Gift Amount: [Donation amount]
- Tax-deductible Amount: [Tax deductible amount]
- This organization's tax ID is: [Tax ID of NPO]
- Tracking Code: [Convio confirmation/transaction code]

TeamRaiser Autoresponders

- Event/Donation Form: [Name of TR Event]
- In Support Of: [Name of participant, team, or event depending]
- Date: [Date of transaction/purchase]
- Amount: [Amount of transaction]
- Tracking Code: [Convio confirmation/transaction code]

eCommerce Autoresponders

- Site Tax ID: [Tax ID of NPO]
- Total Amount: [Amount of purchase]
- Fair Market Value of Items Ordered: [Value entered by admin when configuring product]

- Shipping Charges: [Value entered by admin when configuring product and/or store]
- Total Tax Deductible Amount: [Total Amount - Fair Market Value of Items Ordered]
- Tracking Code: [Convio confirmation/transaction code]

Tribute Autoresponders

- Fund Name: [Name of the Tribute Fund]
- Date: [Date of transaction/purchase]
- Amount: [Amount of transaction]
- Tracking Code: [Convio confirmation/transaction code]

Thank You Pages

The S 120 tag can display property values on Thank You Pages in several categories.

Donation2, TeamRaiser, and Tributes all include:

- Transaction Summary:
- Transaction Date:

Gift Information

- Amount
- Recognition Name:
- Show Gift Amount to Public:
- Personal Note:

Note: By default, "Show Gift Amount to Public" and "Personal Note" are not included in Tributes.

Billing Information

- Title:
- First Name:
- Middle Name:
- Last Name:
- Suffix:
- Street 1:

- Street 2:
- City:
- State/Province:
- ZIP/Postal Code:
- Country:
- Email Address:
- Remember me:

Payment Information

- Payment type:
- Expiration Date:
- Credit Card Number:
- Gift Amount:
- Tax-deductible Amount:
- This organization's tax ID is:
- Tracking Code:

List of S120 Objects

The following is a complete list of objects that are available for the S120 tag. Not all values below are available in every property of an object.

Tip: To find out what S120 values are available in any property in the named object, add `[[S120:help]]` to the source code.

Important: S120:dc:creditCardExpirationDate, S120:dc:creditCardNumber, and S120:dc:creditCardType do not work in autoresponders.

S120:cart:additionalDonation	S120:cart:class
S120:cart:adminCatId	S120:cart:cmd
S120:cart:applID	S120:cart:constituent
S120:cart:cVSHheader	S120:cart:continueCheckingOut
S120:cart:campaignInfo	S120:cart:crossSellComponent
S120:cart:cartSaved	S120:cart:customAmounts
S120:cart:catId	S120:cart:customStrings
S120:cart:catalogPageDefault	S120:cart:description
S120:cart:checkingOut	S120:cart:empty

S120:cart:engine	S120:cart:userId
S120:cart:featuredProductsComponent	S120:cart:valid
S120:cart:frId	S120:dc:additionalGift
S120:cart:homePageDefault	S120:dc:anonymousDonation
S120:cart:id	S120:dc:cVSHHeader
S120:cart:lastPurchase	S120:dc:campaignId
S120:cart:longDescription	S120:dc:campaignName
S120:cart:name	S120:dc:campaignType
S120:cart:orderId	S120:dc:city
S120:cart:pageNumber	S120:dc:class
S120:cart:pageWrapperId	S120:dc:contactInfo
S120:cart:paymentInfo	S120:dc:country
S120:cart:paymentNavContext	S120:dc:creditCardDonation
S120:cart:paymentProcessorCall	S120:dc:creditCardExpirationDate
S120:cart:premiumSelector	*See note above
S120:cart:previewMode	S120:dc:creditCardNumber
S120:cart:productFolder	*See note above
S120:cart:productList	S120:dc:creditCardType
S120:cart:productNameFilter	*See note above
S120:cart:productPriceFilter	S120:dc:discountCode
S120:cart:productTypeFilter	S120:dc:donFormId
S120:cart:proxyId	S120:dc:donationFormName
S120:cart:proxyName	S120:dc:donationInfo
S120:cart:proxyType	S120:dc:donationLevelId
S120:cart:salesTax	S120:dc:donationLevelName
S120:cart:shoppingCart	S120:dc:donationLevelValue
S120:cart:shoppingContext	S120:dc:donorSpecifiedAmount
S120:cart:shoppingEngine	S120:dc:donorAddress
S120:cart:sortOrder	S120:dc:donorCity
S120:cart:storeInfo	S120:dc:donorCountry
S120:cart:supportsBillLater	S120:dc:donorFirstName
S120:cart:upsellComponent	S120:dc:donorLastName
	S120:dc:donorName

S120:dc:donorState	S120:dc:jointDonation
S120:dc:donorStreet1	S120:dc:jointDonorFirstName
S120:dc:donorStreet2	S120:dc:jointDonorFullName
S120:dc:donorStreet3	S120:dc:jointDonorLastName
S120:dc:donorSuffix	S120:dc:jointDonorName
S120:dc:donorTitle	S120:dc:jointDonorTitle
S120:dc:donorZip	S120:dc:lastName
S120:dc:ecardMessage	S120:dc:matchEmail
S120:dc:ecardRecipients	S120:dc:matchFormUrl
S120:dc:ecardSelector	S120:dc:matchInstructions
S120:dc:ecardSendDate	S120:dc:matchPhone
S120:dc:ecardSenderName	S120:dc:membershipGiftMessageBo dy
S120:dc:ecardSubject	S120:dc:membershipGiftMessageCl osing
S120:dc:echeckAccountNumber	S120:dc:membershipGiftMessageSi gnature
S120:dc:echeckRoutingNumber	S120:dc:membershipGiftMessageSu bject
S120:dc:email	S120:dc:membershipRenewal
S120:dc:employer	S120:dc:memorialDonation
S120:dc:eventName	S120:dc:messageToParticipant
S120:dc:firstName	S120:dc:middleName
S120:dc:fullName	S120:dc:name
S120:dc:fundName	S120:dc:newMembership
S120:dc:futurePayments	S120:dc:numberOfItems
S120:dc:giftAidStatus	S120:dc:occupation
S120:dc:giftAmount	S120:dc:optIn
S120:dc:giftType	S120:dc:originalTransAmount
S120:dc:goodsAndServicesValue	S120:dc:otherAmount
S120:dc:honoreeFullName	S120:dc:participantName
S120:dc:honoreeName	S120:dc:paymentMethod
S120:dc:honoreeTitle	S120:dc:paymentMode
S120:dc:includeGiftAmountInNotifica tion	S120:dc:phone
S120:dc:initialPayment	
S120:dc:interestCategories	

S120:dc:pledgeAmount	S120:dc:shippingLastName
S120:dc:postalCode	S120:dc:shippingPostalCode
S120:dc:premiumId	S120:dc:shippingStateProvince
S120:dc:premiumName	S120:dc:shippingStreet1
S120:dc:premiumSelected	S120:dc:shippingStreet2
S120:dc:premiumShippingCharge	S120:dc:shippingTitle
S120:dc:premiumValue	S120:dc:sponsoreeName
S120:dc:professionalSuffix	S120:dc:stateProvince
S120:dc:publicRecognitionName	S120:dc:street1
S120:dc:recognitionName	S120:dc:street2
S120:dc:recurringCurrentPaymentNumber	S120:dc:suffix
S120:dc:recurringCurrentStatusString	S120:dc:taxDeductibleValue
S120:dc:recurringFrequencyLabel	S120:dc:teamCaptainName
S120:dc:recurringLastPaymentDate	S120:dc:teamName
S120:dc:recurringNextPaymentDate	S120:dc:teamRaiser
S120:dc:recurringNumberPaymentsCompleted	S120:dc:teamRaiserId
S120:dc:recurringNumberPaymentsOutstanding	S120:dc:title
S120:dc:recurringTotalAmountPaid	S120:dc:trackingCode
S120:dc:recurringTotalNumberOfPayments	S120:dc:transactionDate
S120:dc:recurringTotalPaymentAmount	S120:dc:transactionID
S120:dc:refundAmount	S120:dc:transactionSummary
S120:dc:rewardPointsEarned	S120:dc:tribute
S120:dc:salutation	S120:dc:tributeDonation
S120:dc:shippingCharge	S120:dc:tributeHonoreeAddress
S120:dc:shippingCity	S120:dc:tributeMessageBody
S120:dc:shippingCountry	S120:dc:tributeMessageClosing
S120:dc:shippingFirstName	S120:dc:tributeMessageIncludeGiftAmount
S120:dc:shippingFullName	S120:dc:tributeMessageSignature
	S120:dc:tributeNotifyRecipientAddress
	S120:dc:tributeNotifyRecipientName
	S120:dc:tributeType

S120:don_rc:autoresponderType

S120:don_rc:cVSHeder

S120:don_rc:class

S120:don_rc:hashCode

S120:don_rc:renderMode

S120:upsell:amountRemaining

S120:upsell:class

S120:upsell:currentAmount

S120:upsell:inPreviewMode

Appendix B: S Tags by Application

Platform

S0	S20	S63	S100	S334
S1	S25	S66	S101	S355
S3	S26	S72	S102	S356
S4	S29	S80	S120	S364
S6	S36	S81	S121	S366
S7	S50	S87	S188	S500
S8	S53	S88	S201	
S9	S55	S97	S202	
S11	S59	S99	S203	

Advocacy

S76
S90
S93
S94
S96
S1151
S1152
S1153

Centers

S83

Communication

S38
S58
S541
S565

Constituent

S103

Contacts

S18

S19

S45

Contributed Content

S67

S68

S69

S79

Directory

S65

Donations

S15

S57

Donations2

S98

Facebook

S61

Navbar

S2

S311

PageBuilder

S22	S27	S51
-----	-----	-----

S24	S39	S151
-----	-----	------

Personal Events

S421

S422

Personal Fundraising

S530

S535

S542

S548
S549

Redirector

S37

Reports

S16
S406

Rewards

S54

StoryBuilder

S24
S32
S52

Survey

S28

TeamRaiser

S35	S48	S347	S402	S407
S42	S330	S348	S403	S408
S43	S335	S349	S404	S413
S44	S345	S350	S405	S414
S47	S346	S400		S415

Tell-A-Friend

S5

UIControls

S550

Widgets

S336
S337

Appendix C: List of Application IDs

The following lists the Application ID that can be used with S4 and S11 and their Java servlet Alias. [The S4 tag will output these values; the S11 tag will take them as an argument.]

1 Donation	30 FrameSet
2 Calendar	31 UserConsole
4 Category	32 ReceiptViewer
5 TellAFriend	33 VoteCenter
6 Ecommerce	34 Rewards
6 Ecommerce (Premium Selector)	35 YahooStoreCallback
7 CommCenter	36 Clubs
8 Ecommerce Checkout	40 WebxRpcServer
9 Donations 2 (User side)	41 Ecard
10 UserLogin	42 MessageViewer
11 Email	43TRSC
12 Find Representatives	48 TRManage
13 Photo Album User	49 BundleEditor
15 User Links	50 CustomCollections
16 R	81 ConsProfileUser
17 Survey	82 ConsInterestsUser
18 News	83 CO
19 PageServer	85 ConsPreferencesUser
20 Directory	86 ChartViewer
21 Guest List	87 InterestSelector
22 Secure Survey	88 QuickReg
23 Secure Page Server	89 GenericFormParser
24 DU	95 News2
25 Ticketing	96 Stories
26 TeamRaiser	98 RedirectHandler
27 TRR	99 EstablishSession
28 TRC	100 Search
29 TRGiftForm	101 DocServer

103 VolunteerMgmtUser	1016 RelatedLinks
150 Advocacy	1017 PageBuilderPreviewPage
151 ZipDistrict	1018 NewsAdmin
181 ReplyDispatcher	1019 PageBuilder
182 EmailReplyHandler	1020 AdminHomePage
183 PreregAuth	1022 OfflineDonationAdmin
184 LoginAs	1023 CalendarPreview
185 ConvioAdminManager	1024 TicketingPreview
186 ServiceCenter	1026 FriendraiserAdmin
190 ReceiptRequest	1027 FriendraiserPreview
192 GigyaLogin	1030 TellAFriendAdmin
200 PageNavigator	1031 ContentRedirector
201 SPageNavigator	1032 DonationPreview
210 GetTogether	1033 ContentUpload
211 GetTogetherSec	1034 RewardsPreview
261 TellFriendOpt	1035 RewardsAdmin
400 LteUser	1036 RewardRedemptionReport
401 LteUserPreview	1037 News2Preview
1001 AdminLogin	1038 ClubsAdmin
1002 ConsFind	1039 StationeryAdmin
1003 GroupAdmin	1040 EcardAdmin
1003 GroupAdmin	1041 Mailmerge
1004 RolesAdmin	1042 TransactionAdmin
1005 ConsList	1043 ReceiptViewerAdmin
1006 Interactions	1044 Donations 2 (Admin side)
1007 Notes	1046 Donations 2 Preview
1008 PremiumAdmin	1050 PageWrapperEditor
1009 DonationAdmin	1051 FolderAdmin
1011 ConsAdd	1052 EcommerceAdmin
1012 CalendarAdmin	1060 ConvioLogin
1013 PhotoAlbumAdmin	1075 AdminReports
1014 SurveyAdmin	1076 Reporter
1015 SupportTools	1077 Reporter

1078 SiteData	1129
1079 NavBarAdmin	OnlineOfflineComparisonReport
1080 ConsProfileAdmin	1130 DonationTenderReport
1081 ConsConfig	1131 ContentArchive
1082 TaskAdmin	1132 TransactionReport
1083 WorkgroupAdmin	1134 ProspectListImportStatsReport
1084 ConsInterestsAdmin	1135
1085 ConsGroupAdmin	ProspectListPerformanceReport
1086 ConsPreferencesAdmin	1136 CampaignTimelineReport
1087 ImageLibrary	1137 CampaignSummaryReport
1088 AdminConsole	1138 CampaignAnalysisReport
1089 XRefTable	1139 MultiPartEventGuestListReport
1090 Exchange	1140 InterestCategory
1092 LinkSelector	1141 SecurityCategory
1093 Wysiwyg	1142 QueryAdmin
1095 News2Admin	1143 BatchPayment
1096 StoriesAdmin	1144 PoliticalContrib
1097 ConditionalSelector	1145 ProspectAdmin
1098 ProductConfig	1146 FileUploadServer
1099 ComponentSelector	1147 PixelServer
1100 RedirectAdmin	1149 SFileUploadServer
1101 DocumentAdmin	1150 AdvocacyAdmin
1102 AdminDocServer	1151 AdvocacyRequeue
1110 AutoresCenter	1152 AlertFailedDeliveryReport
1121 GuestListReport	1153 AlertDeliveryReport
1122 CampaignReport	1154 AlertSummary
1123 DonationHistory	1155 AlertResponseFederal
1124 DonorHistory	1156 AlertStateHouse
1125 CompanyMatchingReport	1157 AlertStateSenate
1126 TributeReport	1158 AlertRespondent
1127 UserReport	1159 AlertRecipient
1128 EventListReport	1160 SurveyReport
	1161 RealObjectsBackend

1162 DonationsReport	1602 TREM
1163 AdvocacyAnalysisReport	1611 TRAPIPreview
1164 EmailAnalysisReport	1701 CenterStandardPageAdmin
1165 InteractionAnalysisReport	1830 LoginAsAdmin
1166 AdvocacyTimelineReport	1900 OrgEventEdit
1167 InteractionTimelineReport	2010 Duplicates
1168 EmailTimelineReport	2020 SettlementReport
1169 FundraisingDashboard	2030 SettlementSummaryReport
1170 AdvocacyDashboard	3100 VoteCenterAdmin
1171 EmailTrackingReport	3200 DiscountCenterAdmin
1200 FriendraiserSummaryReport	3301 PageBuilderAdmin
1201 FriendraiserRegReport	5200 CommCenterUser
1202 FriendraiserPerfReport	
1203	
FriendraiserDivisionSummaryReport	
1204	
FriendraiserTeamSummaryReport	
1205 FriendraiserComparisonReport	
1206 FriendraiserGiftReport	
1207	
FriendraiserCompanyGiftReport	
1210 PremiumReport	
1211 RecurringPaymentReport	
1212	
RecurringExpectedPaymentReport	
1213 RecurringSummaryReport	
1220 DBManager	
1230 AffiliateAdmin	
1250 ChannelSelector	
1260 DataSyncAdmin	
1270 ReportViewer	
1280 DirectoryAdmin	
1351 AdvocacyPreview	
1360 MercAcctSetup	