Oracle Knowledge Information Manager Tag Library Quick Reference

Oracle Knowledge Version 8.4.2.2

Document Number IMTLI84-QR-22

November 6, 2011

Oracle, Inc.

The InQuira Information Manager Tag Library is a collection of custom actions based on the JavaServer Pages 1.1 (JSP) tag extension API. The custom actions (tags) provide JSP developers the convenience of developing Web applications that incorporate sophisticated client-side features and backend integration.

By using the InQuira Information Manager Tag Library, even HTML developers can easily access information stored in the InQuira Information Manager content repository or legacy systems without knowing the JSP APIs involved. Additionally, the InQuira Information Manager custom tags also provide sophisticated yet simple and elegant mechanisms to generate presentation code for desktop browser software, narrow-band wireless devices, and PDAs.

- form.wizardform
- get.securefiletoken
- auto.login
- benchmark
- cache
- cache.destroy
- create.category.data
- dataset.batch
- dataset.batch.page.iterator
- debug
- delete.content.casenumber
- email.send.page
- emailsubscription.renew
- form.channel.contribution
- form.dataform
- form.emailsubscription.subscribe
- form.emailsubscription.unsubscribe
- form.faq.askquestion
- form.login
- form.lostpassword
- form.message
- form.recommendation.contribution
- form.search.attribute
- form.search.fulltext
- form.select.locale
- form.send.email

- iterate.wizard.previous.responses
- iterate.wizardform.fields
- logout
- manage.content
- node.count
- pdf
- save.content.attribute
- save.content.data
- save.user.attribute
- set.content.casenumber
- set.search.term
- set.shoppingcart.status
- set.user.keyvalue
- sitemap
- template.definition
- template.get
- template.put
- timer
- update.content.metric
- user.admin.link
- user.input.remove.file
- get.contenthistory.data
- get.contenthistory.record

* Required parameter

form.wizardform

top

<IM:form.wizardform
success=*"string"*
Page user will be directed to if the wizard encounters no problems.

error=*"string"*
Page user will be directed to if the wizard encounters problems.

target=*"string"*
Target window for success and error pages.

name=*"string"*
HTML form name.

onsubmit=*"string"*
onSubmit event code.

onreset=*"string"*
onreset event code.

wizardid=*"string"*
The id of the Process Wizard that will be used.

wizardstepid=*"string"*
The step id inside of the Process Wizard.

wizardnextstep=*"string"*
For future use.

id=*"string"*
-
The id of the form.

back=*"true" or "false"*
-
Whether or not the user hit the back button.

searchstring=*"string"*
-
The question asked.

baseurl=*"string"*
-
For analytics. To determine the origin of the request.

querysrc*=*"string"*
-

For analytics. To determine the page of the request.

uniqueid=*"string"*
-
For analytics. To determine the page of the request.

></IM:form.wizardform >

Description:
Creates an HTML form to take input from a Process Wizard question.

---

get.securefiletoken

top

<IM:get.securefiletoken

></IM:get.securefiletoken >

Description:
Generates a necessary token value to append to a secure file link. Works with the IMServletFilter feature.

---

auto.login

top

<IM:auto.login
login=*"string"*
-
Login for the user.

password=*"string"*
-
Password for the user.Needed for LDAP or IAuthenticator login.

vars=*"string"*
-

Extra data for IAuthenticator.

recordid=*"string"*
-
Record id for the user.

/>

Description:
Automatically logs in a user by providing a user login or user recordid.

---

benchmark

top

<IM:benchmark

></IM:benchmark >

Description:
Displays the performance metrics of the page.

---

cache

top

<IM:cache
id*=*"string"*
-
ID for this cached piece of HTML.

minutes=*"integer"*
-
Sets this cache entry to expire in a certain number of minutes.

disablelogin=*"true" or "false"*
-

If set to true, this tag caches page data even if there is a logged in user. Default is false.

disablecache=*"true" or "false"*
-
Set to true to programmatically disable this cache. If true the data enclosed by this tag will process normally. Default is false.

idonlykey=*"true" or "false"*
-
Set to true to only use the supplied id as the key for this cache, otherwise the key will be a combination of the id and the page this tag is located. Default is false.

></IM:cache >

Description:
Caches page data for quick retrieval

---

cache.destroy

top

<IM:cache.destroy
id*=*"string"*
-
ID for this cached piece of HTML.

/>

Description:
Kills the specified cache.

---

create.category.data

top

<IM:create.category.data
referencekey*=*"string"*
-

The reference key for the new category. Reference keys must be unique in a Repository

name*=*"string"*
-
Display name for the new category.

description=*"string"*
-
Description of the new category.

parentcategory=*"string"*
-
Reference key for the parent category. If this value is present, the new category will added as a child of parentcategory. The parent category must exist.

/>

Description:
Automatically creates a category.

---

dataset.batch

top

<IM:dataset.batch
id=*"non-scriptable string"*
-
A desired prefix to use for all scripting variables created by this tag. There is no default id. Make sure all ids are unique.

displayontop=*"integer"*
-
If set to true, the display batch will be showed at the first index of the repetition.

displayonbottom=*"integer"*
-
If set to true, the display batch will be showed at the last index of the repetition.

></IM:dataset.batch >

Description:
Calculates and displays the number of batches for a dataset iterarator.

---

dataset.batch.page.iterator

top

<IM:dataset.batch.page.iterator
id=*"non-scriptable string"*
-
A desired prefix to use for all scripting variables created by this tag. There is no default id. Make sure all ids are unique.

></IM:dataset.batch.page.iterator >

Description:
Iterates over the number of pages of the dataset.batch tag.

---

debug

top

<IM:debug
view=*"string"*
-
Scope of the debug information to view (all, session, request, application, or page).

/>

Description:
Display detailed debugging information on all application, session, request and page level variables.

Example Usage:
<IM:debug view="all"/>

Example Usage:
<IM:debug view="all"/>

Scripting Variables:
none

Body Tags:
none

---

delete.content.casenumber

top

<IM:delete.content.casenumber
documentid=*"string"*
-
Delete the case number for the supplied documented.

recordid=*"string"*
-
Delete the case number for the supplied recorded.

casenumber*=*"string"*
-
The case number to be deleted for the content record.

></IM:delete.content.casenumber >

Description:
Deletes a case number for a content record.

---

email.send.page

top

<IM:email.send.page
recipient*=*"string"*
-
Recipient Email address.

recipientname=*"string"*
-
Recipient display name.

from*=*"string"*
-
From address.

fromname=*"string"*
-
From display name.

subject*=*"string"*
-
Subject for email.

comments=*"string"*
-
Comments to be appended. This comment will be inserted in the desired page if and only if the <code>SC_COMMENT_REPLACE</code> is present.

page=*"string"*
-
Page to be sent. This attribute should be in the SiteMap format: i.e. HOME

url=*"string"*
-
Page to be sent. This attribute should be in the http format: i.e. http://www.inquira.com/IM/index?page=home

></IM:email.send.page >

Description:
Automatically sends the specified page or url to the desired email address.

---

emailsubscription.renew

top

<IM:emailsubscription.renew

recordid*=*"string"*

-

The recordid of the subscription being renewed.

></IM:emailsubscription.renew >

Description:
Renew a given subscription for another 90 days.

---

form.channel.contribution

top

<IM:form.channel.contribution
success=*"string"*

-

Page user will be directed to if the content was successfully added.

error=*"string"*

-

Page user will be directed to if creating the content failed.

channel*=*"string"*

-

The channel this contribution is for.

views=*"string"*

-

The department(s) this contribution is for - delimited by '+'.

categories=*"string"*

-

The categories assigned to this contribution - delimited by '+'.

groups=*"string"*

-

The user groups assigned to this contribution - delimited by '+'.

publish=*"true" or "false"*

-

Set this attribute to false if you do not wish the contribution to be published. The default behaviour is to publish the contributions if the specified channel does not have workflow turned on.

name=*"string"*

-

HTML form name.

onsubmit=*"string"*

-

onSubmit event code.

onreset=*"string"*

-

onreset event code.

useeditor=*"true" or "false"*

-

If this attribute is set to true, every schema attribute that is of type Rich Text Editor will display a wysiwyg component. The default value is true.

allowfilebrowsing=*"true" or "false"*

-

If this attribute is set to true and the useeditor attribute is set to true, allowed users will be able to browse and upload files to the server. The default value is true.

casenumber=*"string"*

-

Assign a case number to this record.

casedescription=*"string"*

-

Assign a case description to this record.

caseincident=*"string"*

-

Assign a case incident to this record.

localecode=*"string"*

-

Desired locale code for this contribution. This value is case sensitive and should match the desired locale's code in the database. If the locale is not found or the locale is not part of the repository's available locales an exception will be thrown.

></IM:form.channel.contribution >

Description:
Creates an HTML form to allow an end user to contribute content directly into a specific content channel through the client site. The user will not be able to edit the record once submitted. If workflow is setup for the channel, the new record will go in to the initial step of the workflow and await approval prior to publishing. Content Administrators can find and locate the contributed record just as if it had been created using the administration tool.

Example Usage:
<IM:form.channel.contribution channel="news" departments="department1+department2" success= "thankyou" error="errorpage">
Title:
<IM:input.channel.contribution attribute="/news/title"/>
Body
<IM:input.channel.contribution attribute="/news/body"/>
</IM:channel.contribute.form>

Scripting Variables:
none

Body Tags:
input.channel.contribution

---

form.dataform

top

<IM:form.dataform
success=*"string"*
-
Page user will be directed to if the dataform was successfully saved.

error=*"string"*
-
Page user will be directed to if the dataform failed to be saved.

dataform=*"string"*
-
Reference Key of the dataform.

target=*"string"*

-
Target window for success and error pages.

view=*"string"*
-
Reference Key of the view where the response should be assigned to.

localeCode=*"string"*
-
Requested Locale code e.g. en for English.

name=*"string"*
-
HTML form name.

onsubmit=*"string"*
-
onSubmit event code.

onreset=*"string"*
-
onreset event code.

></IM:form.dataform >

Description:
Creates an HTML form to allow an end user to enter data into a dataform. Note that this tag can be used either stand alone, or as a child of a get.channel.record tag (e.g. when you have a rating associated with a piece of content).
Example Usage:
Displaying the Form

```
<IM:form.dataform dataform="INQUIRIES" success="requestthank" error="requestthank">
   <IM:iterate.dataform.question>
      <IM:get.dataform.question.record>
         <b><IM:get.dataform.question/></b><br>
         <IM:iterate.dataform.answer>
            <IM:get.dataform.answer.record>
               <IM:input.dataform.answer/><br>
            </IM:get.dataform.answer.record>
         </IM:iterate.dataform.answer>
      </IM:get.dataform.question.record>
   </IM:iterate.dataform.question> <br>
   <input type="submit" value="Submit">
</IM:form.dataform>
```

Displaying Results

```
<IM:get.dataform.results name="nameofform" individual="false" aggregate="false">
  <IM:iterate.dataform.question>
     <IM:get.dataform.question.record>
        <b><IM:get.dataform.question/></b><br>
        <IM:iterate.dataform.answer>
           <IM:get.dataform.answer.record>
              <IM:get.dataform.answer/><br>
           </IM:get.dataform.answer.record>
        </IM:iterate.dataform.answer>
     </IM:get.dataform.question.record>
  </IM:iterate.dataform.question>
</IM:get.dataform.results>
```

Scripting Variables:
none

Body Tags:
iterate.dataform.question
get.dataform.question.record
get.dataform.question
get.dataform.answer.record
iterate.dataform.answers
input.dataform.answer

---

form.emailsubscription.subscribe

top

<IM:form.emailsubscription.subscribe
unsubscribe=*"string"*
-
if set to "TRUE", the tag will try to unsubscribe the user from the passed in item and type.

name=*"string"*
-
name of the subscription to create, if not provided it will be generated automatically.

userid=*"string"*
-
The userid for the user who's subscribing or unsubscribing, if not provided the logged in user will be used.

type*=*"string"*
-
The type of subscription this is, Valid types are CHANNEL,CONTENT,FORUM,TOPIC.

recordid*=*"string"*
-
The recordid of the item to subscribe or unsubscribe to. E.g. the contentID, channelID, forumID, or topicID.

categories=*"string"*
-
List of categories separated by plus (+) signs.

viewlocale=*"string"*
-
The value of current locale for article.

></IM:form.emailsubscription.subscribe >

Description:
Subscribe or Unsubscribe to a given record

---

form.emailsubscription.unsubscribe

top

<IM:form.emailsubscription.unsubscribe
success=*"string"*
-
Page user will be directed to if the subscribe action is successful.

error=*"string"*
-
Page user will be directed if the subscribe action is unsuccessful.

name=*"string"*
-
HTML form name.

onsubmit=*"string"*
-

onSubmit event code.

onreset=*"string"*
-
onreset event code.

></IM:form.emailsubscription.unsubscribe >

Description:
Creates an HTML form used for unsubscribing from subscriptions.

---

form.faq.askquestion

top

<IM:form.faq.askquestion
success=*"string"*
-
Page user will be directed to if the question was successfully created.

error=*"string"*
-
Page user will be directed to if adding the question failed.

topic=*"string"*
-
Topic for this faq.

name=*"string"*
-
HTML form name.

onsubmit=*"string"*
-
onSubmit event code.

onreset=*"string"*
-
onreset event code.

></IM:form.faq.askquestion >

Description:
Creates an HTML form to capture new FAQ questions. New FAQ questions are made available in the admin tool where they can be sorted into specific topics.

Example Usage:

```
<IM:iterate.faq.records topic="myfaq">
   <IM:get.faq.record id="currentquestion">
      <b>Q: <IM:get.faq.question/></b> <br/>
      A: <IM:get.faq.answer/><p><br>
   </IM:get.faq.record>
</IM:iterate.faq.records>
```

<IM:form.faq.askquestion success="faqthankyou" error="faqerror">

```
   Question: <br>
   <IM:input.faq.question size="80" maxlength="80" value=""/><br>
   <input type="submit" value="Submit Question"/>
```

</IM:form.faq.askquestion>

Scripting Variables:
none

Body Tags:
input.faq.question

---

form.login

top

<IM:form.login
success=*"string"*
-
Page user will be directed to if the login was successful

error=*"string"*
-
Page user will be directed to if the login failed, the original data entered will be redisplayed if directed back to the login page, error messages are available using the error.xxx tags

target=*"string"*

-
Target window for success and error pages.

name=*"string"*
-
HTML form name

onsubmit=*"string"*
-
onSubmit event code

onreset=*"string"*
-
onreset event code

></IM:form.login >

Description:
Creates an HTML form to capture login in information such as user id and password.

Example Usage:
```
<IM:form.login success="home" error="logon">
   <IM:input.user.id/>
   <IM:input.password/>
</IM:form.login>
```

Scripting Variables:
none

Body Tags:
input.user.id
input.password

---

form.lostpassword

top

<IM:form.lostpassword
success=*"string"*
-
Page user will be directed to if the login was successful..

error=*"string"*
-
Page user will be directed to if the email address failed, the original data entered will be redisplayed if directed back to the lost password page, error messages are available using the error.xxx tags.

name=*"string"*
-
HTML form name.

onsubmit=*"string"*
-
onSubmit event code.

onreset=*"string"*
-
onreset event code.

></IM:form.lostpassword >

Description:
Creates an HTML form to capture the email address of a user to email them their login information

Example Usage:
```
<IM:form.lostpassword success="home" error="logon">
   <IM:error.email/><br/>
   <IM:input.user.email value="" size="40" maxlength="40"/><br/>
   <input type="submit" value="Submit" name="B1">
</IM:form.lostpassword >
```

Scripting Variables:
none

Body Tags:
input.user.email

---

form.message

top

<IM:form.message
success=*"string"*
-

Page user will be directed to if the message was successfully added.

error=*"string"*

-

Page user will be directed to if adding the message failed.

parentrecordid*=*"string"*

-

The record id of the parent record for this message. Examples would be if an article has talkback associated with it, this attribute would have to contain the record id of the article (the talkback will be a child record of the article). Likewise, for a discussion group, this attribute would either contain the record id of the discussion group (top level talkback), or the record id of another talkback (the talkback becomes a child of another talkback aka. a response).

emailto=*"string"*

-

Email address to send this talkbackresponse to. It could be used for sending it to a mailing list.

emailfrom=*"string"*

-

Email address from who this response is being sent. If you whish to let the users input their own from email address, just create a html input field with name = 'emailfrom', and leave this attribute blank. If no from address is supplied the system will use the ADMIN_EMAIL parameter as the from address for the email. NOTE: in order to send responses as emails, there must be a emailto address specified.

emailfooter=*"string"*

-

Text that will be appended on email responses only, right after the text the user entered. This is an optional attribute, and it can be used to providing a link back to the threaded discussion. You could specify the email footer here as a attribute of this tag or provide a input field with name = 'emailfooter' for this parameter.

name=*"string"*

-

HTML form name.

onsubmit=*"string"*

-

onSubmit event code.

onreset=*"string"*

-

onreset event code.

></IM:form.message >

Description:
Create an HTML form to allow users to contribute to a threaded message board.

Example Usage:
Displaying All Messages**:**

```
 <!-- get message data  -->
<IM:get.message.data dataset="data" topic="hammers"/>
<!-- iterate through messages in discussion records  -->
<IM:iterate.message.records dataset="data">
   <!-- get handle to individual record -->
   <IM:get.message.record>
      <!-- space message to show hierarchy and display message title and text-->
      <img src="images\spacer.gif" width="<% =message.depth %>" aligh="left"/><IM:get.message.title/> <br>
      <img src="images\spacer.gif" width="<% =message.depth %>" aligh="left"/><IM:get.message.text/> <br>
      <!-- display a reply link that passes the message record id to the reply page -->
      <a href="index?page=messagereply&rec=<%=message.recordid%>">Reply</a> <br>
   </IM:get.message.record>
</IM:iterate.message.records>
```

Replying to a Message:

```
<!-- declare page variable and load recordid parameter off url from previous page -->
<% String rec = request.getParameter("rec");%>
<!-- create HTML form for input fields -->
<IM:form.message parentrecordid="<%=rec%>" success="successpage" error="errorpage">
   <!-- display input fields for title and text -->
   Title<br>
   <input.message.title><br>
   Text<br>
   <input.message.text><br>
   <!-- display submit button -->
   <input type="submit" value="Submit">
</IM:form.message>
```

Scripting Variables:

Body Tags:
form.message
input.message.title
input.message.text

form.recommendation.contribution

top

<IM:form.recommendation.contribution
success="string"
-
Page user will be directed to if the content was successfully added.

error="string"
-
Page user will be directed to if creating the content failed.

channel="string"
-
The channel this contribution is for.

priority="string"
-
The priority this contribution will have.

localecode="string"
-
The locale for this recommendation.

views="string"
-
The views assigned to this contribution - delimited by '+'.

docid="string"
-
The docid for the content record, if this recommendation is for an existing content record.

contentid="string"
-
The recordid for the content record, if this recommendation is for an existing content record.

categories="string"
-
The categories assigned to this contribution - delimited by '+'.

name="string"
-

HTML form name.

onsubmit=*"string"*
-
onSubmit event code.

onreset=*"string"*
-
onreset event code.

></IM:form.recommendation.contribution >

Description:
Creates an HTML form to allow an end user to contribute a content recommendation.

---

form.search.attribute

top

<IM:form.search.attribute
channel=*"string"*
-
Name (referencekey) for the Channel to be searched.

matchtype=*"string"*
-
determines whether to match all criteria or any criteria. Valid values are 'ALL' or 'ANY'. IMe insensitive.

success=*"string"*
-

error=*"string"*
-

sortattribute=*"string"*
-
DEPRECATED USE sortby. Specifies attribute used to sort results.

sortdirection=*"string"*

-

DEPRECATED USE direction. Specify either ASC (ascending) or DESC (descending) sort order.

categories=*"string"*

-

delimited (+) list of Content Categories (using reference key) to include in the full text search. i.e. CAT1+CAT2.

matchallcategories=*"true" or "false"*

-

Used in conjunction with 'categories' attribute. Set to true to 'and' the categories together in the search, and false to 'or' the categories.

views=*"string"*

-

delimited (+) list of Views (using reference key) to include in the full text search. i.e. VIEW1+VIEW2.

maxrecords=*"integer"*

-

Max amount of records returned by this search. Default is set to 200 records. A value of 0 will return all possible records.

name=*"string"*

-

HTML form name.

onsubmit=*"string"*

-

onSubmit event code.

onreset=*"string"*

-

onreset event code.

sortby=*"string"*

-

Sort By field for the result set. Valid fields: contentchannel, docid, startdate, enddate, masteridentifier.

direction=*"string"*

-

The direction of the sort. Ascending or descending. The default if no value provided will be ascending.

></IM:form.search.attribute >

Description:
Create HTML form to perform attribute search on specified Content Channel.

Example Usage:
<IM:form.search.attribute channel="directory" matchtype="ALL" success="searchresults" error="errorpage">
First Name:
<IM:input.search.attribute attribute="directory\first_name"/>

Last Name:
<IM:input.search.attribute attribute="directory\last_name"/>

<input type="submit"/>
</IM:form.search.attribute>

Scripting Variables:
none

Body Tags:
input.search.attribute

---

form.search.fulltext

top

<IM:form.search.fulltext
channels=*"string"*
-
A delimited (+) list of Content Channels (using reference key) to include in the full text search. i.e. NEWS+EVENTS.

success=*"string"*
-
Page user will be directed to if the search was successfully created.

error=*"string"*
-
Page user will be directed to if search failed.

categories=*"string"*
-

delimited (+) list of Content Categories (using reference key) to include in the full text search. i.e. CAT1+CAT2.

matchallcategories=*"true" or "false"*

-

Used in conjunction with 'categories' attribute. Set to true to 'and' the categories together in the search, and false to 'or' the categories.

matchallterms=*"true" or "false"*

-

If this attribute is set to true, the search will AND all the words searched. If it is set to false, this tag will OR the words searched. The default value is true.

views=*"string"*

-

delimited (+) list of Views (using reference key) to include in the full text search. i.e. VIEW1+VIEW2.

maxrecords=*"integer"*

-

Max amount of records returned by this search. Default is set to 200 records. A value of 0 will return all posible records

name=*"string"*

-

HTML form name

onsubmit=*"string"*

-

onSubmit event code

onreset=*"string"*

-

onreset event code

sortby=*"string"*

-

Sort By field for the results. Valid fields: contentchannel, docid, startdate, enddate, masteridentifier

direction=*"string"*

-

The direction of the sort. Ascending or descending. The default if no value provided will be ascending.

></IM:form.search.fulltext >

Description:
Create an HTML form to perform a full text search on one or more selected Content Channels.

Example Usage:

```
<IM:form.search.fulltext channels="news+events+press" success="searchresults" error="errorpage">
   Search String: <br>
   <IM:input.search.fulltext/>
   <br>
   <input type="submit"/>
</IM:form.search.fulltext>
```

Scripting Variables:
none

Body Tags:
input.search.fulltext

---

form.select.locale

top

<IM:form.select.locale
success=*"string"*
-
Page user will be directed to if the locale change was successful.

error=*"string"*
-
Page user will be directed to if the locale change failed, the original data entered will be redisplayed if directed back to the login page, error messages are available using the error.xxx tags.

name=*"string"*
-
HTML form name.

onsubmit=*"string"*
-
onSubmit event code.

onreset=*"string"*

-
onreset event code.

></IM:form.select.locale >

Description:
Create an HTML form that displays a drop down box of available locals for the current domain.

Example Usage:
```
<IM:form.select.locale  success="successpage" error="errorpage">
    <input type="submit"/>
</IM:form.select.locale>
```

Scripting Variables:
none

Body Tags:
none

---

form.send.email

top

<IM:form.send.email
subject*=*"string"*
-
subject line of the email

page=*"string"*
-
html page to send as the body of the email

url=*"string"*
-
html page to send as the body of the email

success=*"string"*
-
Page user will be directed to if the edit was successful

error=*"string"*

-
Page user will be directed to if the edit failed, the original data entered will be redisplayed if directed back to the edit page, error messages are available using the error.xxx tags.

name=*"string"*
-
HTML form name

onsubmit=*"string"*
-
onSubmit event code

onreset=*"string"*
-
onreset event code

></IM:form.send.email >

Description:
Allows for the sending of a client html page as email.

---

form.shoppingcart.action

top

<IM:form.shoppingcart.action
cart*=*"string"*
-
Refrence key for shopping cart.

options*=*"string"*
-
Name of the HTML element containing the options or attributes of the product.

success=*"string"*
-
Page user will be directed to if the subscribe action is successful.

error=*"string"*
-

Page user will be directed if the subscribe action is unsuccessful.

cartaction*=*"string"*
-
Action to be performed. Valid values for this field are add and <B>remove.

item*=*"string"*
-
Recordid of item to be added or deleted from shopping cart.

transactionid=*"string"*
-
Optional field for setting the transactionid if known. This attribute will get the transaction directly from the database instead of the user's session, or cookie.

name=*"string"*
-
HTML form name

onsubmit=*"string"*
-
onSubmit event code

onreset=*"string"*
-
onreset event code

></IM:form.shoppingcart.action >

Description:
Creates an HTML form used for adding or removing items from a shopping cart.

---

form.shoppingcart.update

top

<IM:form.shoppingcart.update
cart*=*"string"*
-
Refrence key for shopping cart.

transactionid=*"string"*
-
Optional field for setting the transactionid if known. This attribute will get the transaction directly from the database instead of the user's session, or cookie.

success=*"string"*
-
Page user will be directed to if the update action is successful

error=*"string"*
-
Page user will be directed if the update action is unsuccessful

name=*"string"*
-
HTML form name

onsubmit=*"string"*
-
onSubmit event code

onreset=*"string"*
-
onreset event code

></IM:form.shoppingcart.update >

Description:
Creates an HTML form used for updating items from a shopping cart.

---

form.subscription.find.email

top

<IM:form.subscription.find.email
success=*"string"*
-
Page user will be directed to if the email search is successful

error="string"
-
Page user will be directed if the email search is unsuccessful

name="string"
-
HTML form name

onsubmit="string"
-
onSubmit event code

onreset="string"
-
onreset event code

></IM:form.subscription.find.email >

Description:
Creates an HTML form used for selecting an email address to be used for searching subscriptions assigned to it. Typically used in the subscribe/unsubscribe process.

---

form.subscription.subscribe

top

<IM:form.subscription.subscribe
success="string"
-
Page user will be directed to if the subscribe action is successful

error="string"
-
Page user will be directed if the subscribe action is unsuccessful

name="string"
-
HTML form name

onsubmit="string"

-
onSubmit event code

onreset=*"string"*
-
onreset event code

></IM:form.subscription.subscribe >

Description:
Creates an HTML form used for subscribing and unsubscribing from subscriptions.

---

form.user.attributes

top

<IM:form.user.attributes
success=*"string"*
-
Page user will be directed to if the edit was successful

error=*"string"*
-
Page user will be directed to if the edit failed, the original data entered will be redisplayed if directed back to the edit page, error messages are available using the error.xxx tags

createnew=*"true" or "false"*
-
Set to <code>true</code> if you would like to create a new user. Otherwise set to <code>false</code>. The default behavior is set to <code>false</code>

adminuser=*"true" or "false"*
-
Set to <code>true</code> if you would like to create a new admin user. Set to <code>false</code> to create Web Users. The default behavior is set to create web users

autologin=*"true" or "false"*
-
Set to <code>true</code> if you would like to login the edited user. Otherwise set to <code>false</code>. The default behavior is set to <code>false</code>

userid=*"string"*

-

User id (guid) of the user to be edited

roles=*"string"*

-

When creating new users, use this attribute to specify one or more Roles to be assigned to the new user. Use existing role reference keys separated by plus signs. If no roles are specified, then the default user role will be assigned to the new user.

parenteditorgroup=*"string"*

-

If this attribute is present, a new Editor Group will be created using this attribute as its Parent Brach (make sure the branch exists). The Editor Group creation feature is only applied for new users. The new Editor Group will have as reference key the new user's login and its Display Name format will be: user's Last Name, user's First Name.

name=*"string"*

-

HTML form name

onsubmit=*"string"*

-

onSubmit event code

onreset=*"string"*

-

onreset event code

useeditor=*"true" or "false"*

-

If this attribute is set to true, every schema attribute that is of type Rich Text Editor will display a wysiwyg component. The default value is true.

allowfilebrowsing=*"true" or "false"*

-

If this attribute is set to true and the useeditor attribute is set to true, allowed users will be able to browse and upload files to the server. The default value is true.

></IM:form.user.attributes >

Description:
Creates an HTML form to allow a user to edit their profile information.

Example Usage:

```
<IM:form.user.attributes  success="successpage" error="errorpage">
   User ID: <IM:input.user.id.error/><br>
   <IM:input.user.id value="" system size="20"/><br>
   Password: <IM:input.user.password .error/><br>
   <IM:input.user.password value="" size="20"/><br>
   First Name: <IM:input.user.firstname .error/><br>
   <IM:input.firstname value="" size="30"/><br>
   Last Name: <IM:input.user.lastname .error/><br>
   <IM:input.user.lastname value="" size="30"/><br>
   Email Address: <IM:input.user.email.error/><br>
   <IM:input.user.email value="" size="30"/><br>
   Phone Number:<br>
   <IM:input.user.attribute attribute="demo/phone" value="" size="10"/><br>
   Zip Code<br>
   <IM:input.user.attribute attribute="demo/zipcode" value="" size="5"/><br>
   <br>
   <IM:input.submit value="Save Changes"/>
</IM:form.user.attributes>
```

Scripting Variables:

none

Body Tags:

input.user.id
input.user.password
input.user.firstname
input.user.lastname
input.user.email
input.user.attribute

---

get.browsesearch.data

top

<IM:get.browsesearch.data
dataset*=*"string"*
-
The name of the dataset being created. Used by the scripting variables

direction=*"string"*
-

ASCENDING or DESCENDING. Default is DESCENDING

sortby=*"string"*

-

Default value is MODIFIED_TIME. Only one type can be sorted by per request. Multiple values are invalid. The list of valid values is guid (IM guid), title (document title), facet ('|' separated list of facets), status (IM status – 'published' or 'draft'), mdt (modification timestamp in ms since Jan 1, 1970), fmdt (formatted modification timestamp), excerpt (document excerpt), extId (IM doc id), uniqueId (search unique id), colId (collection id), docId (doc id), colName (collectionName), lang (language), enc (character encoding), docType (document type), url (display url)

startdate=*"string"*

-

Specifies the start date for documents in either milliseconds or a formatted date in either %m/%d/%Y or %m/%d/%Y %H:%M:%S formats

enddate=*"string"*

-

Specifies the end date for documents in either milliseconds or a formatted date in either %m/%d/%Y or %m/%d/%Y %H:%M:%S formats

maxrecords*=*"string"*

-

This parameter will be the maximum number of results to return

channels=*"string"*

-

Specify this parameter to restrict results to a channel or a plus-delimited list of channels

categories=*"string"*

-

Specify this parameter to restrict results to a category or a plus-delimited list of categories

views=*"string"*

-

Specify this parameter to restrict results to a view or a plus-delimited list of views

usergroups=*"string"*

-

Specify this parameter to restrict results to a usergroup or a plus-delimited list of usergroups

findiml=*"string"*

-

Specifies the data you want to search for - i.e. the question

sortobjects=*"string"*

-

A delimited list of objects that the search results will be sorted by. The format is object1+object2+object3. Valid values for the sort objects are: guid (IM guid), title (document title), facet ('|' separated list of facets), status (IM status – 'published' or 'draft'), mdt (modification timestamp in ms since Jan 1, 1970), fmdt (formatted modification timestamp), excerpt (document excerpt), extId (IM doc id), uniqueId (search unique id), colId (collection id), docId (doc id), colName (collectionName), lang (language), enc (character encoding), docType (document type), url (display url), Direct IML expression

params=*"string"*

-

A delimited list of extra name value pairs that are passed into the FindDocuments search request. These parameters are of the format name=value+name=value

chunksize=*"string"*

-

Defines the number of records to process during the validation of the IM documents. The default value is 200 records at a time.

validate=*"true" or "false"*

-

A boolean flag that is used to determine whether to check if the IM doc ids are still valid before returning the search results back to the UI. This validation can add a lot of time if the result set is large. The default value is TRUE

baseurl=*"string"*

-

For analytics. To determine the origination of the request

detailobjects=*"string"*

-

The columns we are expecting to be returned. EXT_ID + this value, comma separated. The list of valid values are: guid (IM guid), title (document title), facet ('|' separated list of facets), status (IM status – 'published' or 'draft'), mdt (modification timestamp in ms since Jan 1, 1970), fmdt (formatted modification timestamp), excerpt (document excerpt), extId (IM doc id), uniqueId (search unique id), colId (collection id), docId (doc id), colName (collectionName), lang (language), enc (character encoding), docType (document type), url (display url)

outputformat=*"string"*

-

The format we are expecting to be returned. Valid values are COMMA_SEPARATED_LIST, documentList, documentRecord. Default is COMMA_SEPARATED_LIST. Output format documentRecord will use detailobjects attribute to tell search which columns to return.

imdocs=*"true" or "false"*
-
Flag to determine whether to only return IM documents. Default is true.

id*=*"string"*
-
Unique id for this tag

pagesize=*"integer"*
-
The number of results you would like at a time.

pagestart=*"integer"*
-
The number of results to start from. For example, if you have pagesize set to 20, the second call you would set the page start to 20 so it would return items at 21 and beyond.

querysrc*=*"string"*
-
The page making the call. This attribute is required for analytic reports!!

></IM:get.browsesearch.data >

Description:
Tag that submits a request to the InQuira search engine and retrieves a collection of content guids.

---

get.browsesearch.record

top

<IM:get.browsesearch.record
id*=*"string"*
-
unique id for record to retrieve

></IM:get.browsesearch.record >

Description:
Tag that submits a request to the InQuira search engine and retrieves a collection of content guids.

get.category.attribute

top

<IM:get.category.attribute
attribute*=*"string"*
-
Name of attribute that should be displayed.

 Allowed Attribute values:
 name -- returns the name of the category
 referencekey -- returns the reference key of the category
 recordid -- returns the recordid(guid) of the category
 description -- returns the description of the category
 sortorder -- returns the sort order of the category within the level it resides in
 level -- returns the depth level of the category
 parentreferencekey -- returns the reference key of the parent of this category
 parentid -- returns the recordid(guid) of the parent of this category

></IM:get.category.attribute >

Description:
Tag that displays the selected attribute for a category. This tag must be included inside a get.category.record tag.

get.category.data

top

<IM:get.category.data
dataset*=*"string"*
-
A unique identifier for the dataset. This identifier is used by the iterate.records tag to access records that are retrieved.

view=*"string"*
-
List of views delimited by a + sign that the ad-hoc query will search on. This means that will return category parent records that have assigned the passed in views.

channel=*"string"*
-

List of channels separated b y a + sign that the category query will search on. This means that will return categories parent records that have assigned the passed in channels

discussion=*"string"*

-

Discussion Board Reference key. If this attribute is present, the tag will return a list of parent categories associated with a discussion board.

discussionid=*"string"*

-

Discussion Board ID. If this attribute is present, the tag will return a list of parent categories associated with a discussion board.

sortby=*"string"*

-

This parameter determines which category parameter will be used for sorting. Allowed values are: name, referencekey, and sortOrder.

direction=*"string"*

-

The direction of the sort. Ascending or descending. The default if no value provided will be ascending.

recordid=*"string"*

-

ID of the parent category to retrieve. If you provide a value for record id, this tag will return the children for the category referenced by that value

referencekey=*"string"*

-

Reference Key of the parent category record to retrieve. If you provide a value for referencekey, this tag will return the children for the category referenced by that value

localecode=*"string"*

-

Locale code for the display name.

/>

Description:

Retrieves a list of categories. If neither channels nor views are specified, this tag will return all parent categories for the active domain. If views and/or channels are specified, this tag will return parent categories that are assigned to those values passed in. The views and channel assignment are performed on the SiteConnect Management Console.

Scripting Variables: - count = total number of category objects retrieved.

get.category.record

top

<IM:get.category.record
recordid=*"string"*
-
ID of record to retrieve.

referencekey=*"string"*
-
Reference Key of category record to retrieve.

id=*"non-scriptable string"*
-
A desired prefix to use for all scripting variables created by this tag. There is no default id. Make sure all ids are unique

localecode=*"string"*
-
Locale code for the display name.

></IM:get.category.record >

Description:
Tag that gets a category record either from iteration or by a referencekey.

get.channel.attribute

top

<IM:get.channel.attribute
attribute*=*"string"*
-
Case sensitive xpath to the desired content attribute.

id=*"non-scriptable string"*

-
A desired prefix to use for all scripting variables created by this tag. The default value is 'attribute'. If nesting channel-attribute tags, make sure all ids are unique

processbody=*"true" or "false"*
-
set to 'true' if you create a body for this tag

type=*"string"*
-
type of the attribute for formatting purposes, can be number, date, or text (default is text)

mask=*"string"*
-
Used in conjunction with the type attribute, a mask is used if you want to explicitly change how a date or number is formatted. Use a mask for example if a number should be displayed as currency (i.e. $1.00 or $1 instead of the '1' the value would come back as).

parentinlaw=
-
TODO

></IM:get.channel.attribute >

Description:
Get a specific attribute from the current content record. To simply display an attribute the tag does not need a body section. To access the scripting variables however, set processbody to true and optionally assign an id to the tag.

Formatting Dates:
The default mask pattern for a date attribute type is `%Y-%m-%d %H:%M:%S %Z`. You can specify a pattern using the mask parameter with the get.channel.attribute tag in conjunction with the type parameter. For dates formatting, that type parameter must be set to "date". When the mask parameter is used to convert a date to a string, it uses the following pattern specifiers:

Conversion Specifiers

Specifier
Description

`%%`
a `'%'` character

`%a`

abbreviated weekday name

`%A`
full weekday name

`%b`
abbreviated month name

`%B`
full month name

`%c`
shorthand for "`%X %x`", the locale format for date and time

`%d`
day of the month as a decimal number (01-31)

`%e`
same as `%d` but does not print the leading 0 for days 1 through 9

`%F`
milliseconds as a decimal number (000-999)

`%H`
hour based on a 24-hour clock as a decimal number (00-23)

`%I`
hour based on a 12-hour clock as a decimal number (01-12)

`%j`
day of the year as a decimal number (001-366)

`%m`
month as a decimal number (01-12)

`%M`
minute as a decimal number (00-59)

`%p`
AM/PM designation for the locale

`%S`

second as a decimal number (00-59)

`%w`
weekday as a decimal number (0-6), where Sunday is 0

`%x`
date using the date representation for the locale

`%X`
time using the time representation for the locale

`%y`
year without century (00-99)

`%Y`
year with century (such as 1990)

`%Z`
time zone name (such as Pacific Daylight Time)

`%z`
time zone name (such as Pacific Daylight Time)

Formatting Numbers:
You can also specify a pattern for numeric values using the mask parameter with the get.channel.attribute tag in conjunction with the type tag. For numeric formatting, the type tag must be set to "number". When the mask parameter is used to convert a numeric value to a string, pattern strings can include the following types of characters:

- Numbers

  Pattern strings can include numeric characters. Wherever a number is included in a pattern string, the number is displayed unless an input character in the same relative position overwrites it. For example, suppose for the positive pattern string `"9,990.00"`, and the attribute value is 53.88 to which the pattern has been applied. The resulting display for the value is 9,953.88.

- Separators

  Pattern strings can include the period character (`.`) as a decimal separator, and comma character (`,`) as a thousand separator.

- Placeholders

  You use the pound sign character (#) to represent numeric characters that will be displayed to the user. For example, for the positive pattern `"$#,##0.00"`, if the attribute value is 76329, it would be displayed as $76,329.00. Strictly speaking, however, you don't need to use placeholders. The format strings `",0.00"`, `"#,#0.00"`, and `"#,##0.00"` are functionally equivalent. In other words, including

separator characters in a pattern string signals the mask formatter to use the separators, regardless of whether you use (or where you put) placeholders. The placeholder character's chief virtue lies in its ability to make pattern strings more human-readable.

- Spaces

  To include a space in a pattern string, use the underscore character (_). This character inserts a space if no numeric character has been input to occupy that position.

- Currency

  The dollar sign character ($) is the canonical currency mark in pattern strings.

All other characters specified in a pattern string are displayed as typed. The following table shows examples of the how the value 1019.55 is displayed for different positive patterns:

Pattern String	Display
"#,##0.00"	1,019.55
"$#,##0.00"	$1,019.55
"___,__0.00"	1,019.55

Example Usage:

```
<!--Get channel data records using query-->
<IM:get.channel.data query="topnews" dataset="mynews">
<!-- Iterate channel record -->
<IM:iterate.channel.record dataset="mynews">
   <!--Get handle to current record-->
   <IM:get.channel.record id="news">
      <!-- Display title attribute for current record -->
      <IM:get.channel.attribute attribute="NEWS\TITLE">
      <IM:get.channel.attribute attribute="NEWS\IMAGE" processbody="true" id="image">
         <%if (image.exists) {%>
           <img src="<%=news.resourcepath%>/<%=image.value%>">
         <%}%>
      </IM:get.channel.attribute>
    </IM:get.channel.record>
</IM:iterate.channel.record>
```

Scripting Variables:
{id}.exist **-** Does data exist in the attribute
{id}.value - Value of current attribute

Body Tags:
none

---

get.channel.attribute.exists

top

<IM:get.channel.attribute.exists
negate=*"non-scriptable string"*
-
Reverses the conditional of this tag. If it evaluates to "true" it will show the contents of this tag if the value is not present. The default behavior of this attribute is to be false.

attribute*=*"string"*
-
The desired attribute to check.

></IM:get.channel.attribute.exists >

Description:
Checks if the attribute enter has a value set for the current record. If it does exists it displays the contents of this tag. If negate is set to true, this tag will display its contents only if the value does not exists.

---

get.channel.attribute.value

top

<IM:get.channel.attribute.value
attribute*=*"string"*
-
Case sensitive xpath to the desired content attribute.

id=*"non-scriptable string"*
-
A desired prefix to use for all scripting variables created by this tag. The default value is 'attribute'. If nesting channel-attribute tags, make sure all ids are unique

processbody=*"true" or "false"*
-
set to 'true' if you create a body for this tag

type=*"string"*
-
type of the attribute for formatting purposes, can be number, date, or text (default is text)

mask=*"string"*
-
Used in conjunction with the type attribute, a mask is used if you want to explicitly change how a date or number is formatted. Use a mask for example if a number should be displayed as currency (i.e. $1.00 or $1 instead of the '1' the value would come back as).

parentinlaw=
-
TODO

></IM:get.channel.attribute.value >

Description:
Get a specific attribute value as scripting variables from the current content record.

Formatting Dates:
The default mask pattern for a date attribute type is `%Y-%m-%d %H:%M:%S %Z`. You can specify a pattern using the mask parameter with the get.channel.attribute tag in conjunction with the type parameter. For dates formatting, that type parameter must be set to "date". When the mask parameter is used to convert a date to a string, it uses the following pattern specifiers:

Conversion Specifiers

Specifier
Description

`%%`
a `'%'` character

`%a`
abbreviated weekday name

`%A`
full weekday name

`%b`
abbreviated month name

`%B`
full month name

`%c`
shorthand for "`%X %x`", the locale format for date and time

`%d`
day of the month as a decimal number (01-31)

`%e`
same as `%d` but does not print the leading 0 for days 1 through 9

`%F`
milliseconds as a decimal number (000-999)

`%H`
hour based on a 24-hour clock as a decimal number (00-23)

`%I`
hour based on a 12-hour clock as a decimal number (01-12)

`%j`
day of the year as a decimal number (001-366)

`%m`
month as a decimal number (01-12)

`%M`
minute as a decimal number (00-59)

`%p`
AM/PM designation for the locale

`%S`
second as a decimal number (00-59)

%w
weekday as a decimal number (0-6), where Sunday is 0

%x
date using the date representation for the locale

%X
time using the time representation for the locale

%y
year without century (00-99)

%Y
year with century (such as 1990)

%Z
time zone name (such as Pacific Daylight Time)

%z
time zone name (such as Pacific Daylight Time)

Formatting Numbers:
You can also specify a pattern for numeric values using the mask parameter with the get.channel.attribute tag in conjunction with the type tag. For numeric formatting, the type tag must be set to "number". When the mask parameter is used to convert a numeric value to a string, pattern strings can include the following types of characters:

- Numbers

  Pattern strings can include numeric characters. Wherever a number is included in a pattern string, the number is displayed unless an input character in the same relative position overwrites it. For example, suppose for the positive pattern string "9,990.00", and the attribute value is 53.88 to which the pattern has been applied. The resulting display for the value is 9,953.88.

- Separators

  Pattern strings can include the period character (.) as a decimal separator, and comma character (,) as a thousand separator.

- Placeholders

  You use the pound sign character (#) to represent numeric characters that will be displayed to the user. For example, for the positive pattern "$#,##0.00", if the attribute value is 76329, it would be displayed as $76,329.00. Strictly speaking, however, you don't need to use placeholders. The format strings ",0.00", "#,#0.00", and "#,##0.00" are functionally equivalent. In other words, including separator characters in a pattern string signals the mask formatter to use the separators, regardless of whether you use (or where you put) placeholders. The placeholder character's chief virtue lies in its

ability to make pattern strings more human-readable.

- Spaces

  To include a space in a pattern string, use the underscore character (_). This character inserts a space if no numeric character has been input to occupy that position.

- Currency

  The dollar sign character ($) is the canonical currency mark in pattern strings.

All other characters specified in a pattern string are displayed as typed. The following table shows examples of the how the value 1019.55 is displayed for different positive patterns:

Pattern String	Display
"#,##0.00"	1,019.55
"$#,##0.00"	$1,019.55
"___,__0.00"	1,019.55

Example Usage:

```
<!--Get channel data records using query-->
<IM:get.channel.data query="topnews" dataset="mynews">
<!-- Iterate channel record -->
<IM:iterate.channel.record dataset="mynews">
   <!--Get handle to current record-->
   <IM:get.channel.record id="news">
      <!-- Display title attribute for current record -->
      <IM:get.channel.attribute attribute="NEWS\TITLE">
      <IM:get.channel.attribute attribute="NEWS\IMAGE" processbody="true" id="image">
         <%if (image.exists) {%>
           <img src="<%=news.resourcepath%>/<%=image.value%>">
         <%}%>
      </IM:get.channel.attribute>
   </IM:get.channel.record>
</IM:iterate.channel.record>
```

Scripting Variables:
{id}.exist - Does data exist in the attribute
{id}.value - Value of current attribute

Body Tags:
none

get.channel.data

top

<IM:get.channel.data
features=*"string"*
-
List of features delimited by '+' sign to be added to xml stream. Valid values are : categories, departments, security. Default is all features off.

dataset*=*"string"*
-
A unique identifier for the dataset, this identifier is used by the iterate.channel.records and get.channel.template tags to access records that are retrieved.

query=*"string"*
-
Reference key name of the stored query defined in the Management Console

channel=*"string"*
-
Content Channel containing the data to be displayed. Do not use in when query is specified.

startdate=*"string"*
-
start date of the content.

enddate=*"string"*
-
End date of the content.

mode=*"string"*
-
validitydates - content must be valid at specified time, startdates - content must start by specified date.

expression=*"string"*

-

Expression representing date range.

departments=*"string"*

-

Deprecated. Replace by views.

views=*"string"*

-

List of views delimited by a + sign that the ad-hoc query will search on, only used if the query attribute is not used.

categories=*"string"*

-

List of topics delimited by a + sign that the query will search on.

editorgroups=*"string"*

-

List of editorgroups delimited by a + sign that the query will search on.

usergroups=*"string"*

-

List of usergroups delimited by a + sign that the ad-hoc query will search on, only used if the query attribute is not used.

where=*"string"*

-

Sets the where clause to be used on this query.

sortby=*"string"*

-

List of XPaths delimited by a + sign for sorting the results.

direction=*"string"*

-

The direction of the sort. Ascending or descending. The default if no value provided will be ascending.

matchallcategories=*"true" or "false"*

-

Used in conjunction with 'category' attribute. Set to true to 'and' the topics together in query, and false to 'or' the topics.

hierarchicalcategories=*"true" or "false"*

-

Used in conjunction with 'category' attribute. Set to true to use hierarchical categories, false otherwise. Default is true

ignoredisplaydates=*"true" or "false"*

-

Set to <code>true</code> to fetch all content regardless of Display Start and Display End Date restrictions. The default behavior is not to ignore these dates.

maxrecords=*"integer"*

-

Maximum number of records to retrieve.

updatedsince=*"string"*

-

Retrieve content that has been updated after this date.

postalcode=*"string"*

-

If the passed in channel has GeoSpatial capabilities, fetches records with the passed in postal code and within the proximity radius.

proximity=*"string"*

-

If the passed in channel has GeoSpatial capabilities, sets the proximity radius.

ownerid=*"string"*

-

If the ownerid (login name) is passed in, the records that are fetched are restricted to the specified owner.

casenumber=*"string"*

-

If the casenumber is passed in, the records that are fetched are restricted to the specified casenumber.

/>

Description:

Retrieve valid records from a content channel by defining selection criteria or by using a named query. Data is assigned to a named variable using the dataset parameter. Using a name query is the easiest method for retrieving content channel records. Selection criteria can also be defined at runtime when necessary to retrieve specific records based on some user action.

The CAS administration tool provides an easy to use point and click query builder. Using this tool, any number of queries can be defined and called using the get.channel.data tag. To retrieve records from a content channel using a predefined query, simply use the "query" parameter and provide the reference key for the query.

To dynamically define selection criteria at runtime, the get.channel.data tag provides a number of additional parameters. These parameters can also be used to override criteria of a named query. Dynamic criteria selection parameters includes: channel, departments, categories, startdate, enddate.

To retrieve all valid records from a channel simply specify the channel name using the channel parameter. To restrict the data set to records assigned to specific departments, use the department parameter to specify one or more departments. To specify more then one department use a "+" character between departments reference keys. To restrict the data set to records tagged with specific categories use the category parameter to specify one or more categories, again using the "+" character to delimit. The matchallcategories parameter specifies if all defined categories need to match for the record to be retrieved.

To set or modify the date range of the data set, use the startdate and enddate parameters. Note that these parameters are relative to the publish date of a content record. The remove date of a content record is used to restrict access to records after the current date surpasses the end date.

To limit the total number of records retrieved use the maxrecords parameter.

Example Usage:
Get data for a named query:
<IM:get.channel.data query="todaysnews"/>

Get data for using dynamic selection criteria:
<IM:get.channel.data channel="news"/>

Get data for using dynamic selection criteria specifying specific departments:
<IM:get.channel.data channel="news" departments="HR+IS+SALES" />

Get data for using dynamic selection criteria specifying specific dates to get news for one day only:
<IM:get.channel.data channel="news" startdate="08/31/2002" enddate="08/31/2002" />

Scripting Variables:
{id}.count **-** Number of records retrieved

Body Tags:
none

---

get.channel.record

top

<IM:get.channel.record
features=*"string"*

-
List of features delimited by '+' sign to be added to xml stream. Valid values are : categories, departments, security. Default is all features off.

recordid=*"string"*
-
ID of record to retrieve.

documentid=*"string"*
-
Document ID of record to retrieve. This value overrides the recordid value

id=*"non-scriptable string"*
-
A desired prefix to use for all scripting variables created by this tag. There is no default id. If nesting channel-item tags, make sure all ids are unique

activitytype=*"string"*
-
Activity Identifier to be stored along with the user activity logs. This can be helpful to identify which pages the user visits.

localecode=*"string"*
-
Locale code for the current request. Locale code is of format en_US where en=language, US=location. Default is default locale of repository

usergroups=*"string"*
-
List of usergroups delimited by a + sign that the ad-hoc query will search on, only used if the query attribute is not used.

impressions=*"true" or "false"*
-
Attribute to turn or off the impressions for this record. The default value is to increment the impressions. Set to false not to increment impressions

userviews=*"true" or "false"*
-
Attribute to turn on or off the fetching of content viewed by the logged in user. The default value is set to false to increase performance.

uservisits=*"true" or "false"*

-
Attribute to turn on or off the recording of user visits. The default value is set to false to increase performance.

securedbyview=*"true" or "false"*
-
Attribute to turn on or off Views security. If no values are passed in the views parameter, the security will be based on the actual views of the logged in user. Otherwise the security will be applied based on the passed in views

views=*"string"*
-
List of views delimited by a + sign that will be applied to views security

includedraft=*"true" or "false"*
-
retrieve record including draft version

></IM:get.channel.record >

Description:
Get a handle to a specific channel record using a recordid. If used within the body of an iterate.channel.records tag no parameters are required because the iterator will automatically pass in the current record. The id parameter defined the namespace for scripting variables within the body of the tag.

Example Usage:
Stand alone:

```
<!-- Get passed in record id form URL-->
<% String rec = request.getParameter("rec");%>

<!--Get handle to current record-->
<IM:get.channel.record recordid="<%=rec%>">
   <!-- Display title and body attribute for current record -->
   <IM.get.channel.attribute attribute="NEWS\TITLE"><br>
   <IM.get.channel.attribute attribute="NEWS\BODY">
</IM:get.channel.record>
```

Within an Iterator:

```
<!--Get channel data records using query-->
<IM:get.channel.data query="topnews" dataset="mynews">
<!-- Iterate channel record -->
<IM:iterate.channel.record dataset="mynews">
   <!--Get handle to current record-->
   <IM:get.channel.record id="news">
```

```
    <!-- Display title attribute for current record -->
    <IM.get.channel.attribute attribute="NEWS\TITLE">
    <!-- Provide hyperlink to detail page for current record -->
    <a href="index?page=detail&rec=<%=news.recordid%>>Read more</a><br>
  </IM:get.channel.record>
</IM:iterate.channel.record>
```

Scripting Variables:
{id}.index - Current record index number
{id}.recordid - ID of current record

Body Tags:
get.channel.attribute

---

get.channel.record.exists

top

<IM:get.channel.record.exists
recordid=*"string"*
-
The record id of the channel record to be checked.

documentid=*"string"*
-
The documentid id of the channel record to be checked.

negate=*"non-scriptable string"*
-
Reverses the conditional of this tag. If it evaluates to "true" it will show the contents of this tag if the value is not present. The default behavior of this attribute is to be false.

></IM:get.channel.record.exists >

Description:
Checks if the channel record specified exist or is still valid.

---

get.channel.recordid

top

<IM:get.channel.recordid

></IM:get.channel.recordid >

Description:
Prints to the file the recordid attribute of the current channel record.

---

get.channel.template

top

<IM:get.channel.template
dataset=*"string"*
-
Name of the <channel-data> recordset that contains the data to be transformed, this name corresponds to the id attribute of the channel-data tag, and is IMe sensitive

template*=*"string"*
-
The reference key (not name) of the predefined XSL template that will be used to transform the data retrieved using the channel-data tag

/>

Description:
Formats a channel dataset using a specified XSL stylesheet.

<IM:channel-format dataset="xmldata" template="NewsStyleSheet"/>

This will get content data out of the xmldata variable and transform it using the NewsStyleSheet.

---

get.config.param.value

top

<IM:get.config.param.value
id="string"
-
ID for this record.

propertyName*="string"
-
property name to retrieve value of. This is displayed in the config.properties file.

></IM:get.config.param.value >

Description:
Gets the management console configuration parameter for this repository.

---

get.content.data

top

<IM:get.content.data
dataset*="string"
-
A unique identifier for the dataset. This identifier is used by the iterate.records tag to access records that are retrieved.

channel*="string"
-
List of channels separated by plus(+) signs

localecode="string"
-
Locale to retrieve the records for e.g. fr_FR or en_GB

views="string"
-
List of views separated by plus(+) signs

usergroups="string"
-
List of user groups separated by plus(+)s signs

categories=*"string"*
-
List of categories separated by plus(+)s signs

editorgroups=*"string"*
-
List of editor groups separated by plus(+)s signs

startdate=*"string"*
-
start date attribute of content record

enddate=*"string"*
-
end date attribute of content record

mode=*"string"*
-
validitydates - content must be valid at specified time, startdates - content must start by specified date.

maxrecords=
-
Maximum number of records to retrieve.

sortby=*"string"*
-
List of sort parameters separated b y a + sign.

direction=*"string"*
-
The direction of the sort. Ascending or descending. The default if no value provided will be ascending.

hierarchicalcategories=*"true" or "false"*
-
Used in conjunction with 'category' attribute. Set to true to use hierarchical categories, false otherwise. Default is true

ignoredisplaydates=*"true" or "false"*
-
Set to <code>true</code> to fetch all content regardless of Display Start and Display End Date restrictions. The default behavior is not to ignore these dates.

mostpopular=*"true" or "false"*

-

If this attribute is set to true, to query will return the most popular records only. The sortby attribute will be disabled, however you can specify a direction of the sort.

impressions=*"true" or "false"*

-

If this attribute is set to true, to query will return the impression count for each record.

ownerid=*"string"*

-

If the ownerid (login name) is passed in, the records that are fetched are restricted to the specified owner.

casenumber=*"string"*

-

If the casenumber is passed in, the records that are fetched are restricted to the specified casenumber.

newcontent=*"true" or "false"*

-

Set to true to return records that the active user has not seen yet. This will only apply if there is a logged in user.

reverseHierarchy=*"true" or "false"*

-

Set to true to return records associated with the parents of the passed in categories. Default is false.

viewedby=*"string"*

-

Return records that the passed in userid has already seen. If you would like to sort by the date seen use VIEWEDDATE in the sort parameters.

stagingonly=*"true" or "false"*

-

If this attribute is set to true, to query will return staging documents regardless if they are published or not.

matchallcategories=*"true" or "false"*

-

Used in conjunction with 'category' attribute. Set to true to 'and' the categories together in query, and false to 'or' the categories.

usecache=*"true" or "false"*

-

Set to true to cache the result set. This option is only for published documents not for staging documents

cachetime=*"integer"*
-
If usecache is set to true, this parameter specifies the time in minutes the cache results will live. The default is set to 1 minute

adaptivequery=*"true" or "false"*
-
Set to false to turn off adaptive query. By default is set to true. This parameter optimizes queries order by startdate, enddate, mostpopular, eventstartdate, eventenddate and createdate. If maxrecords is set to 0, adaptive query will be turned off since all records will be returned. To reset it use cache=refresh on the url.

/>

Description:
Retrieves a list of content records using a JDBC Connection pool. This tag looks similar to the get.channel.data tag but all the underlying structure is totally different.

---

get.content.locales

top

<IM:get.content.locales
contentid=*"string"*
-
Recordid of content record to find localized versions of

includedraft=*"true" or "false"*
-
include draft docs when count locales

documentid=*"string"*
-
Document id of content record to find localized versions of

dataset*=*"string"*
-
Name of a local variable used to store the resulting array of localized record references

/>

Description:
Retrieves all locales for a specified content record

---

get.content.record

top

<IM:get.content.record
id=*"non-scriptable string"*
-
A desired prefix to use for all scripting variables created by this tag. There is no default id. Make sure all ids are unique

userviews=*"true" or "false"*
-
Attribute to turn on or off the fetching of content viewed by the logged in user. The default value is set to false to increase performance.

></IM:get.content.record >

Description:
Tag that gets a content record either from an iteration

---

get.dataform.answer

top

<IM:get.dataform.answer

/>

Description:
Retrieves the answer for the current data form answer record.

Example Usage:
Displaying the Form

```
<IM:form.dataform dataform="INQUIRIES" success="requestthank" error="requestthank">
   <IM:iterate.dataform.question>
```

```
      <IM:get.dataform.question.record>
        <b><IM:get.dataform.question/></b><br>
        <IM:iterate.dataform.answer>
          <IM:get.dataform.answer.record>
            <IM:input.dataform.answer/><br>
          </IM:get.dataform.answer.record>
        </IM:iterate.dataform.answer>
      </IM:get.dataform.question.record>
   </IM:iterate.dataform.question> <br>
   <input type="submit" value="Submit">
</IM:form.dataform>
```

Displaying Results

```
<IM:get.dataform.results name="nameofform" individual="false" aggregate="false">
  <IM:iterate.dataform.question>
      <IM:get.dataform.question.record>
        <b><IM:get.dataform.question/></b><br>
        <IM:iterate.dataform.answer>
          <IM:get.dataform.answer.record>
            <IM:get.dataform.answer/><br>
          </IM:get.dataform.answer.record>
        </IM:iterate.dataform.answer>
      </IM:get.dataform.question.record>
   </IM:iterate.dataform.question>
</IM:get.dataform.results>
```

Scripting Variables:
none

Body Tags:
none

---

get.dataform.answer.record

top

```
<IM:get.dataform.answer.record


></IM:get.dataform.answer.record >
```

Description:
Provides a handle to the current data form answer record.

Example Usage:
Displaying the Form

```
<IM:form.dataform dataform="INQUIRIES" success="requestthank" error="requestthank">
   <IM:iterate.dataform.question>
      <IM:get.dataform.question.record>
         <b><IM:get.dataform.question/></b><br>
         <IM:iterate.dataform.answer>
            <IM:get.dataform.answer.record>
               <IM:input.dataform.answer/><br>
            </IM:get.dataform.answer.record>
         </IM:iterate.dataform.answer>
      </IM:get.dataform.question.record>
   </IM:iterate.dataform.question> <br>
   <input type="submit" value="Submit">
</IM:form.dataform>
```

Displaying Results

```
<IM:get.dataform.results name="nameofform" individual="false" aggregate="false">
  <IM:iterate.dataform.question>
      <IM:get.dataform.question.record>
         <b><IM:get.dataform.question/></b><br>
         <IM:iterate.dataform.answer>
            <IM:get.dataform.answer.record>
               <IM:get.dataform.answer/><br>
            </IM:get.dataform.answer.record>
         </IM:iterate.dataform.answer>
      </IM:get.dataform.question.record>
   </IM:iterate.dataform.question>
</IM:get.dataform.results>
```

Scripting Variables:
{id}.index **-** Current record index number
{id}.recordid - ID of current record

Body Tags:
get.dataform.answer

---

get.dataform.name

top

<IM:get.dataform.name

referencekey*=*"string"*

-

The reference key of the desired Data Form.

></IM:get.dataform.name >

Description:
Returns the localized name of a data form passing in a reference key.

---

get.dataform.question

top

<IM:get.dataform.question

/>

Description:
Returns the question for the current data form question record.

Example Usage:
Displaying the Form

```
<IM:form.dataform dataform="INQUIRIES" success="requestthank" error="requestthank">
   <IM:iterate.dataform.question>
      <IM:get.dataform.question.record>
         <b><IM:get.dataform.question/></b><br>
         <IM:iterate.dataform.answer>
            <IM:get.dataform.answer.record>
               <IM:input.dataform.answer/><br>
            </IM:get.dataform.answer.record>
         </IM:iterate.dataform.answer>
      </IM:get.dataform.question.record>
   </IM:iterate.dataform.question> <br>
   <input type="submit" value="Submit">
</IM:form.dataform>
```

Displaying Results

```
<IM:get.dataform.results name="nameofform" individual="false" aggregate="false">
  <IM:iterate.dataform.question>
      <IM:get.dataform.question.record>
         <b><IM:get.dataform.question/></b><br>
         <IM:iterate.dataform.answer>
```

```
            <IM:get.dataform.answer.record>
               <IM:get.dataform.answer/><br>
            </IM:get.dataform.answer.record>
         </IM:iterate.dataform.answer>
      </IM:get.dataform.question.record>
   </IM:iterate.dataform.question>
</IM:get.dataform.results>
```

Scripting Variables:
none

Body Tags:
none

---

get.dataform.question.record

top

<IM:get.dataform.question.record

></IM:get.dataform.question.record >

Description:
Provides a handle to the current data form question record.

Example Usage:
Displaying the Form

```
<IM:form.dataform dataform="INQUIRIES" success="requestthank" error="requestthank">
   <IM:iterate.dataform.question>
      <IM:get.dataform.question.record>
         <b><IM:get.dataform.question/></b><br>
         <IM:iterate.dataform.answer>
            <IM:get.dataform.answer.record>
               <IM:input.dataform.answer/><br>
            </IM:get.dataform.answer.record>
         </IM:iterate.dataform.answer>
      </IM:get.dataform.question.record>
   </IM:iterate.dataform.question> <br>
   <input type="submit" value="Submit">
</IM:form.dataform>
```

Displaying Results

```
<IM:get.dataform.results name="nameofform" individual="false" aggregate="false">
  <IM:iterate.dataform.question>
    <IM:get.dataform.question.record>
      <b><IM:get.dataform.question/></b><br>
      <IM:iterate.dataform.answer>
        <IM:get.dataform.answer.record>
          <IM:get.dataform.answer/><br>
        </IM:get.dataform.answer.record>
      </IM:iterate.dataform.answer>
    </IM:get.dataform.question.record>
  </IM:iterate.dataform.question>
</IM:get.dataform.results>
```

Scripting Variables:
{id}.index - Current record index number
{id}.recordid - ID of current record

Body Tags:
get.dataform.question
iterate.dataform.answer
get.dataform.answer.record
get.dataform.answer
input.dataform.answer

---

get.dataform.results

top

<IM:get.dataform.results
dataform=*"string"*
-
Name of the survey

individual=*"true" or "false"*
-
set to 'true' if you want to get individual results

aggregate=*"true" or "false"*
-
set to 'true' if you want to get aggregate results

></IM:get.dataform.results >

Description:
Retrieves results of a specific dataform.

Example Usage:
Displaying the Form

```
<IM:form.dataform dataform="INQUIRIES" success="requestthank" error="requestthank">
   <IM:iterate.dataform.question>
      <IM:get.dataform.question.record>
         <b><IM:get.dataform.question/></b><br>
         <IM:iterate.dataform.answer>
            <IM:get.dataform.answer.record>
               <IM:input.dataform.answer/><br>
            </IM:get.dataform.answer.record>
         </IM:iterate.dataform.answer>
      </IM:get.dataform.question.record>
   </IM:iterate.dataform.question> <br>
   <input type="submit" value="Submit">
</IM:form.dataform>
```

Displaying Results

```
<IM:get.dataform.results dataform="nameofform" individual="false" aggregate="false">
  <IM:iterate.dataform.question>
      <IM:get.dataform.question.record>
         <b><IM:get.dataform.question/></b><br>
         <IM:iterate.dataform.answer>
            <IM:get.dataform.answer.record>
               <IM:get.dataform.answer/><br>
            </IM:get.dataform.answer.record>
         </IM:iterate.dataform.answer>
      </IM:get.dataform.question.record>
   </IM:iterate.dataform.question>
</IM:get.dataform.results>
```

Scripting Variables:
none

Body Tags:
iterate.dataform.question
get.dataform.question.record
get.dataform.question
get.dataform.answer.record
iterate.dataform.answers

s

get.dataset.batch.page

top

<IM:get.dataset.batch.page
id="non-scriptable string"
-
A desired prefix to use for all scripting variables created by this tag. There is no default id. Make sure all ids are unique

></IM:get.dataset.batch.page >

Description:
Gets the current page from the batch page iterator

---

get.domain.attribute

top

<IM:get.domain.attribute
attribute="string"
-
xpath of extended site attribute to return. Note that extended site properties are defined via schema in the admin tool.

systemattribute="string"
-
Basic site property value to return. Valid values are : <ol> domain - domain name of site refkey - locale neutral reference key for site

view="string"
-
Reference key for the repository view to return property for

/>

Description:
Get a system or custom domain attribute. Only one of the attribute parameters can be specified at one time.

Example Usage:

```
<html>
<head>
   <meta content="<IM:get.domain.attribute attribute="demo/keywords" />" name="KEYWORDS">
   <title><IM:get.domain.attributes systemattribute="domain"/></title>
</head>
<body>
   ...
</body>
</html>
```

Scripting Variables:
none

Body Tags:
none

---

get.emailsubscription.name

top

<IM:get.emailsubscription.name

></IM:get.emailsubscription.name >

Description:
Tag that displays the name of the current subscription. This tag is intended to be a child of get.emailsubscription.record tag

---

get.emailsubscription.record

top

<IM:get.emailsubscription.record
id=*"non-scriptable string"*
-

A desired prefix to use for all scripting variables created by this tag. There is no default id. Make sure all ids are unique

></IM:get.emailsubscription.record >

Description:
Tag that gets the current subscription in the iteration.

---

get.inquirasearch.answer

top

<IM:get.inquirasearch.answer
id*=*"string"*
-
The unique id representing the InquiraResult

></IM:get.inquirasearch.answer >

Description:
Tag that receives the current InquiraResult object from inside the iterate.dataset tag.

---

get.inquirasearch.answerfacet

top

<IM:get.inquirasearch.answerfacet
id*=*"string"*
-
The unique id representing the AnswerFacet

></IM:get.inquirasearch.answerfacet >

Description:
Tag that receives the current AnswerFaccet object from inside the iterate.dataset tag.

---

get.inquirasearch.data

top

<IM:get.inquirasearch.data
type*=*"string"*
-
The type of request to perform: empty|narrow|search|forward|backward|more|open|feedback . More is used to display a great number of facets. If a facet is listed and its "hasMore" property is set to true, you can submit a "more" request type to get the full list of its subfacets.

searchstring=*"string"*
-
When the type of request is "search", the searchstring will be used.

facettoggle=*"string"*
-
Toggle a facet on or off from its current filter state.

dataset*=*"string"*
-
A unique identifier for the dataset, this identifier is used by the iterate.dataset to step through all of the returned answers and facets. To retrieve the facets or answers simply append a ".facets" or ".answers" to the value passed in.

pageobj=*"string"*
-
Future-not used now

classlevel0=*"string"*
-
Have the formatted answers that are returned by the answer.getExcerpt method use a specified css class instead of the default "sippetLevel0"

classlevel1=*"string"*
-
Have the formatted answers that are returned by the answer.getExcerpt method use a specified css class instead of the default "sippetLevel1"

classlevel2=*"string"*
-

Have the formatted answers that are returned by the answer.getExcerpt method use a specified css class instead of the default "sippetLevel2"

classlevel3=*"string"*

-

Have the formatted answers that are returned by the answer.getExcerpt method use a specified css class instead of the default "sippetLevel3"

restrict=*"string"*

-

Tell the InQuira search to only look for InfoManager documents, Discussions or Non-Discussions. The three valid values are IM, IM_DISCUSSION, and IM_CHANNEL respectively. If this parameter is not specified, no restrictions will be applied.

parsedHTML=*"string"*

-

An "open" request will return the html into this variable. The html will contain highlighted text.

iqaction=*"string"*

-

CValue returned with an answer that is used for a subsequent 'open' document request.

highlightinfo=*"string"*

-

Information needed to create highlighting for an "open" request". Part of a returned answer.

answerid=*"string"*

-

A unique value for an answer. Used when calling the request to "open" an external document. Returned for each answer

relatedids=*"string"*

-

A unique value for an answers related ids. Used when calling the request to find "similar" answers.

url=*"string"*

-

The url that the target document resides at

id*=*"string"*

-

The url that the target document resides at

feedbackcomments=*"string"*

-

Comments regarding how well the search has performed. Use feedbackcomments or feedbackratings

feedbackrating=*"string"*

-

A value from 1 to 5 indicating the rating of the search results, where 5 is the best rating. Use feedbackrating or feedbackcomments

segment=*"string"*

-

A value passed into the SOAP request

escalate=*"string"*

-

Used for a RespondContact search request type. Indicates whether or not to escalate based on user satisfaction

params=*"string"*

-

A + delimited way of passing in extra params along with the SOAP request

views=*"string"*

-

A + delimited way of passing in views to restrict by

validate=*"true" or "false"*

-

Vallidate the search results against the database. Default is false

ccaanswersolutionslist=*"string"*

-

used by an AddSolution request

ccaextsollist=*"string"*

-

used by an AddSolution request

ccasrkey=*"string"*

-

used by an AddSolution request

ccatypes=*"string"*

-
used by an AddSolution request

baseurl=*"string"*
-
For analytics. To determine the origination of the request

querysrc=*"string"*
-
For analytics. To determine the page of the request

uniqueid=*"string"*
-
For analytics. To determine the page of the request

facetcollectionid=*"string"*
-
Allows completely different navigation taxonomies

domaingroup=*"string"*
-
Allows completely different navigation taxonomies

questiontype=*"string"*
-
Associated with the questiontype(subject) and language to determine a domaingroup by search engine

wizardlabel=*"string"*
-
For analytics. To determine the page of the request

stepuuid=*"string"*
-
For analytics. To determine the page of the request

wizardid=*"string"*
-
For analytics. To determine the page of the request

wizardstepid=*"string"*
-
For analytics. To determine the page of the request

localecode=*"string"*
-
Desired locale code for this search. If not valid or an exception occurs, the session's locale will be used.

resultlanguage=*"string"*
-
Used as part of the constraint for the result filtering.

></IM:get.inquirasearch.data >

Description:
Tag that submits a request to the InQuira search engine.

---

get.inquirasearch.facet

top

<IM:get.inquirasearch.facet
id*=*"string"*
-
The unique id representing the ResultFacet

></IM:get.inquirasearch.facet >

Description:
Tag that receives the current ResultFacet object from inside the iterate.dataset tag.

---

get.inquirasearch.portlet

top

<IM:get.inquirasearch.portlet
id*=*"string"*
-
The unique id representing the Portlet.

></IM:get.inquirasearch.portlet >

Description:
Tag that receives the current Portlet object from inside the iterate.dataset tag.

---

get.inquirasearch.portlet.item

top

<IM:get.inquirasearch.portlet.item
id*=*"string"*
-
The unique id representing the Portlet's InquiraResult

></IM:get.inquirasearch.portlet.item >

Description:
Tag that receives the current Portlet's InquiraResult object from inside the iterate.dataset tag.

---

get.inquirasearch.tablecell

top

<IM:get.inquirasearch.tablecell
id*=*"string"*
-
The unique id representing the Table Cell

></IM:get.inquirasearch.tablecell >

Description:
Tag that receives the current Table Cell object from inside the iterate.dataset tag.

---

get.inquirasearch.tablerow

top

<IM:get.inquirasearch.tablerow
id*=*"string"*
-
The unique id representing the TableRow

></IM:get.inquirasearch.tablerow >

Description:
Tag that receives the current Table Row object from inside the iterate.dataset tag.

---

get.language.selection

top

<IM:get.language.selection
width=*"integer"*
-
The width to decide the number of cells in one row.

selectedStr=*"string"*
-
The string to initialize the check box, which is seperated by comma.

></IM:get.language.selection >

Description:
Tag retrieves the available languages to display in HTML TABLE.

---

get.locale.attribute

top

<IM:get.locale.attribute
attribute*=*"string"*
-

Name of attribute that should be displayed.

 Allowed Attribute values:
 encoding -- returns the character encoding of the locale
 dateformat -- returns the date format string of the locale used for storing the data
 recordid -- returns the recordid(guid) of the locale
 dateformatdisplay -- returns the date format display string for displaying the locale dates
 timeformat -- returns the time format string of the locale used for storing the data
 timeformatdisplay -- returns the time format string of the locale used for storing the data
 description -- the localized description of the locale description
 code -- the locale code for the locale

></IM:get.locale.attribute >

Description:
Tag that displays the selected attribute for a locale. this tag must be included inside a get.locale.record tag.

---

get.locale.record

top

<IM:get.locale.record
recordid=*"string"*
-
Recordid of content record to find localized versions of

id=*"non-scriptable string"*
-
Name of a local variable used to store the resulting array of localized record references

></IM:get.locale.record >

Description:
Retrieves locale information for a locale

---

get.localized.name

top

<IM:get.localized.name
id=*"non-scriptable string"*
-

A desired prefix to use for all scripting variables created by this tag. The default value is 'localizedname'. If nesting channel-attribute tags, make sure all ids are unique

referencekey*=*"string"*
-
The reference key of the desired object.

type*=*"string"*
-
The type of the desired object. Valid options are: CHANNEL, VIEW, ROLE, CATEGORY, EDITORGROUP, USERGROUP, DATALIST, NEWSLETTER, SHOPPINGCART, DATAFORM

></IM:get.localized.name >

Description:
Returns the localized name of an object passing in a reference key and a type.

---

get.localized.name.value

top

<IM:get.localized.name.value
id=*"non-scriptable string"*
-
A desired prefix to use for all scripting variables created by this tag. The default value is 'localizedname'. If nesting channel-attribute tags, make sure all ids are unique

referencekey*=*"string"*
-
The reference key of the desired object.

type*=*"string"*
-
The type of the desired object. Valid options are: CHANNEL, VIEW, ROLE, CATEGORY, EDITORGROUP, USERGROUP, DATALIST, NEWSLETTER, SHOPPINGCART, DATAFORM

></IM:get.localized.name.value >

Description:
Returns the localized name of an object passing in a reference key and a type as a scripting variable.

get.localized.text

top

<IM:get.localized.text
key*=*"string"*
-
Localized resource key to get the string for as defined in LocaleStrings properties file under the resource directory

parameters=
-
The parameters to make the String not be truncated.

localeCode=*"string"*
-
Locale code for the current request. Locale code is of format en_US where en=language, US=location. Default is default locale of repository

/>

Description:
Displays localized string for the specified resource key from the resource text file. Used to localize static site text and images. The resource text file is located on the jsp server in the WEB-INF\classes directory. Create a file for each local supported by the site, appending the local code at the end of the filename.

ApplicationResources.properties - Default language resource file (if no local is specified by the browser).
ApplicationResources_en.properties - English file defined by "_en"
ApplicationResources_es.properties - Spanish file defined by "_es"

Example Usage:
<IM:get.localized.text key="welcome.text"/>
<img src="<IM:get.localized.text key="welcome.image"/>">

Scripting Variables:
none

Body Tags:
none

get.management.console.url

top

<IM:get.management.console.url

></IM:get.management.console.url >

Description:
Prints the management console url stored in the configuration parameters for this repository.

---

get.management.console.url.value

top

<IM:get.management.console.url.value
id=*"string"*
-
ID for this record.

></IM:get.management.console.url.value >

Description:
Gets the management console url stored in the configuration parameters for this repository.

---

get.message.author

top

<IM:get.message.author

/>

Description:
Get message body text from a message board.

Example Usage:
Displaying All Messages:

```
 <!-- get message data  -->
<IM:get.message.data dataset="data" topic="hammers"/>
<!-- iterate through messages in discussion records  -->
<IM:iterate.message.records dataset="data">
   <!-- get handle to individual record -->
   <IM:get.message.record>
      <!-- space message to show hierarchy and display message title and text-->
      <img src="images\spacer.gif" width="<% =message.depth %>" aligh="left"/><IM:get.message.title/> <br>
      <img src="images\spacer.gif" width="<% =message.depth %>" aligh="left"/><IM:get.message.text/>
<br>
      <!-- display a reply link that passes the message record id to the reply page -->
      <a href="index?page=messagereply&rec=<%=message.recordid%>">Reply</a> <br>
   </IM:get.message.record>
</IM:iterate.message.records>
```

Replying to a Message:

```
<!-- declare page variable and load recordid parameter off url from previous page -->
<% String rec = request.getParameter("rec");%>
<!-- create HTML form for input fields -->
<IM:form.message parentrecordid="<%=rec%>" success="successpage" error="errorpage">
   <!-- display input fields for title and text -->
   Title<br>
   <input.message.title><br>
   Text<br>
   <input.message.text><br>
   <!-- display submit button -->
   <input type="submit" value="Submit">
</IM:form.message>
```

Scripting Variables:
none

Body Tags:
none

---

get.message.data

top

```
<IM:get.message.data
dataset*="string"
-
```

A unique identifier for the dataset, this identifier is used by the channel-iterator and channel-format tags to access records that are retrieved.

topic=*"string"*
-
Reference key name of the stored query defined in the Management Console

/>

Description:
Retrieve published messages for a specific topic and save the result set in a user defined variable. If no topic is defined all topics will be returned.

Example Usage:
Displaying All Messages:

```
 <!-- get message data  -->
<IM:get.message.data dataset="data" topic="hammers"/>
<!-- iterate through messages in discussion records  -->
<IM:iterate.message.records dataset="data">
   <!-- get handle to individual record -->
   <IM:get.message.record>
      <!-- space message to show hierarchy and display message title and text-->
      <img src="images\spacer.gif" width="<% =message.depth %>" aligh="left"/><IM:get.message.title/> <br>
      <img src="images\spacer.gif" width="<% =message.depth %>" aligh="left"/><IM:get.message.text/> <br>
      <!-- display a reply link that passes the message record id to the reply page -->
      <a href="index?page=messagereply&rec=<%=message.recordid%>">Reply</a> <br>
   </IM:get.message.record>
</IM:iterate.message.records>
```

Replying to a Message:

```
<!-- declare page variable and load recordid parameter off url from previous page -->
<% String rec = request.getParameter("rec");%>
<!-- create HTML form for input fields -->
<IM:form.message parentrecordid="<%=rec%>" success="successpage" error="errorpage">
   <!-- display input fields for title and text -->
   Title<br>
   <input.message.title><br>
   Text<br>
   <input.message.text><br>
   <!-- display submit button -->
   <input type="submit" value="Submit">
</IM:form.message>
```

Scripting Variables:
{id}.count - Total records retrieved

Body Tags:
iterate.message.records

get.message.record

top

<IM:get.message.record
recordid=
-
Content Text record id of record to display. Useful for a detailed display page where a user would select a link to a record and be directed to a page that displays the record.

id=
-
Prefix for all scripting variables within the scope of this tag.

></IM:get.message.record >

Description:
Get handle to current message record.

Example Usage:
Displaying All Messages:

```
 <!-- get message data  -->
<IM:get.message.data dataset="data" topic="hammers"/>
<!-- iterate through messages in discussion records  -->
<IM:iterate.message.records dataset="data">
   <!-- get handle to individual record -->
   <IM:get.message.record>
      <!-- space message to show hierarchy and display message title and text-->
      <img src="images\spacer.gif" width="<% =message.depth %>" aligh="left"/><IM:get.message.title/> <br>
      <img src="images\spacer.gif" width="<% =message.depth %>" aligh="left"/><IM:get.message.text/> <br>
      <!-- display a reply link that passes the message record id to the reply page -->
      <a href="index?page=messagereply&rec=<%=message.recordid%>">Reply</a> <br>
   </IM:get.message.record>
</IM:iterate.message.records>
```

Replying to a Message:

```
<!-- declare page variable and load recordid parameter off url from previous page -->
<% String rec = request.getParameter("rec");%>
<!-- create HTML form for input fields -->
<IM:form.message parentid="<%=rec%>" success="successpage" error="errorpage">
   <!-- display input fields for title and text -->
   Title<br>
   <input.message.title><br>
   Text<br>
   <input.message.text><br>
   <!-- display submit button -->
   <input type="submit" value="Submit">
</IM:form.message>
```

Scripting Variables:
{id}.index **-** Current record index number
{id}.recordid - ID of current record
{id}.depth - Message hierarchy depth

Body Tags:
get.message.title
get.message.text
form.message
input.message.title
input.message.text

---

get.message.text

top

```
<IM:get.message.text


/>
```

Description:
Get message body text from a message board.

Example Usage:
Displaying All Messages:

```
 <!-- get message data  -->
<IM:get.message.data dataset="data" topic="hammers"/>
```

```
<!-- iterate through messages in discussion records  -->
<IM:iterate.message.records dataset="data">
   <!-- get handle to individual record -->
   <IM:get.message.record>
      <!-- space message to show hierarchy and display message title and text-->
      <img src="images\spacer.gif" width="<% =message.depth %>" aligh="left"/><IM:get.message.title/> <br>
      <img src="images\spacer.gif" width="<% =message.depth %>" aligh="left"/><IM:get.message.text/>
<br>
      <!-- display a reply link that passes the message record id to the reply page -->
      <a href="index?page=messagereply&rec=<%=message.recordid%>">Reply</a> <br>
   </IM:get.message.record>
</IM:iterate.message.records>
```

Replying to a Message:

```
<!-- declare page variable and load recordid parameter off url from previous page -->
<% String rec = request.getParameter("rec");%>
<!-- create HTML form for input fields -->
<IM:form.message parentrecordid="<%=rec%>" success="successpage" error="errorpage">
   <!-- display input fields for title and text -->
   Title<br>
   <input.message.title><br>
   Text<br>
   <input.message.text><br>
   <!-- display submit button -->
   <input type="submit" value="Submit">
</IM:form.message>
```

Scripting Variables:
none

Body Tags:
none

---

get.message.title

top

```
<IM:get.message.title

/>
```

Description:
Get message title from a record in a message board.

Example Usage:
Displaying All Messages:

```
 <!-- get message data  -->
<IM:get.message.data dataset="data" topic="hammers"/>
<!-- iterate through messages in discussion records  -->
<IM:iterate.message.records dataset="data">
   <!-- get handle to individual record -->
   <IM:get.message.record>
      <!-- space message to show hierarchy and display message title and text-->
      <img src="images\spacer.gif" width="<% =message.depth %>" aligh="left"/><IM:get.message.title/>
<br>
      <img src="images\spacer.gif" width="<% =message.depth %>" aligh="left"/><IM:get.message.text/> <br>
      <!-- display a reply link that passes the message record id to the reply page -->
      <a href="index?page=messagereply&rec=<%=message.recordid%>">Reply</a> <br>
   </IM:get.message.record>
</IM:iterate.message.records>
```

Replying to a Message:

```
<!-- declare page variable and load recordid parameter off url from previous page -->
<% String rec = request.getParameter("rec");%>
<!-- create HTML form for input fields -->
<IM:form.message parentrecordid="<%=rec%>" success="successpage" error="errorpage">
   <!-- display input fields for title and text -->
   Title<br>
   <input.message.title><br>
   Text<br>
   <input.message.text><br>
   <!-- display submit button -->
   <input type="submit" value="Submit">
</IM:form.message>
```

Scripting Variables:
none

Body Tags:
none

---

get.meta.resourcepath.content

top

<IM:get.meta.resourcepath.content

></IM:get.meta.resourcepath.content >

Description:
Prints to the file the meta resource path for the current content record.

---

get.metadata.attribute

top

<IM:get.metadata.attribute
attribute*=*"string"*
-
The attribute you want to get.

></IM:get.metadata.attribute >

Description:
Returns the specified attribute of a metadata object.

---

get.metadata.attribute.value

top

<IM:get.metadata.attribute.value
attribute*=*"string"*
-
The attribute you want to get.

id*=*"string"*
-
ID for this record.

></IM:get.metadata.attribute.value >

Description:
Returns the specified attribute of a metadata object.

---

get.metadata.channels

top

<IM:get.metadata.channels
dataset*=*"string"*
-
A unique identifier for the dataset, this identifier is used by the iterate.channel.records and get.channel.template tags to access records that are retrieved.

registered*=*"true" or "false"*
-
Set to true if you would like to get registered channels.

view=*"string"*
-
The view reference key for registered content channels.

usecache=*"true" or "false"*
-
Set to true to cache the result set.

cachetime=*"integer"*
-
If usecache is set to true, this parameter specifies the time in minutes the cache results will live. The default is set to 1 minute

></IM:get.metadata.channels >

Description:
Gets a list of channels for the specified Domain. If the registered value evaluates to "true" it returns the list of registered channels for the passed in view.

---

get.metadata.dataform

top

<IM:get.metadata.dataform

dataset*=*"string"*

-

A unique identifier for the dataset, this identifier is used by the iterate.channel.records and get.channel.template tags to access records that are retrieved.

view=*"string"*

-

The view reference key for registered content channels.

parentviews=*"true" or "false"*

-

If set to true, it will fetch all Data Forms assigned to the passed in View and also all data forms assigned to all its parents in the tree.

></IM:get.metadata.dataform >

Description:
Gets a list of data forms assigned to pass in view. If no view is passed in it will get the data forms assigned to the Domain.

---

get.metadata.record

top

<IM:get.metadata.record
id*=*"string"*

-

ID for this record.

type=*"string"*

-

IF not in an iteration, passed in the type of the metadata. Valid values are: channel or dataform

referencekey=*"string"*

-

IF not in an iteration, and type is set, this will be the reference key of the meta data.

></IM:get.metadata.record >

Description:
Tag that gets the meta data object. If inside an iteration it will return the current value. If stand alone. you will

need to pass in a type and a reference key. Scripting variables available:
index = current repetition index
recordid = record id of the object
hasdiscussion = if record is of type ContentChannel, this is a boolean value to determine if the channel has discussions turned on
hasratings = if record is of type ContentChannel, this is a boolean value to determine if the channel has rating turned on

---

get.page.error.data

top

<IM:get.page.error.data
id*=*"non-scriptable string"*
-
A desired prefix to use for all scripting variables created by this tag. There is no default id. Make sure all ids are unique

></IM:get.page.error.data >

Description:
Tag that returns an error code that might have occurred in a page. Examples, Not authorize to view topic, topic not published etc...

---

get.privileges

top

<IM:get.privileges
negate=*"non-scriptable string"*
-
Reverses the conditional of this tag. If it evaluates to "true" it will show the contents of this tag. The default behavior of this attribute is to be false.

></IM:get.privileges >

Description:
Gets the privilege for the current content record

get.privileges.add

top

<IM:get.privileges.add
negate=*"non-scriptable string"*
-
Reverses the conditional of this tag. If it evaluates to "true", if the user have privileges, it will show the contents of this tag. The default behavior of this attribute is to be false.

channel*=*"string"*
-
The channel reference key.

view=*"string"*
-
The view reference key.

editorgroups=*"string"*
-
List of editorgroups delimited by a + sign that the user needs to have in order to add a record. This tag will allow a user to add a record if he/she has at least one of the passed in editor groups. In order to make this feature active, the passed in channel needs to have Editor Groups management turned on.

></IM:get.privileges.add >

Description:
Gets the ADD content privilege for the logged in user and for the given channel and view. If no View is specified, it will assume the Domain level .

get.privileges.value

top

<IM:get.privileges.value
negate=*"non-scriptable string"*
-

Reverses the conditional of this tag. If it evaluates to "true", if the user have privileges, it will show the contents of this tag. The default behavior of this attribute is to be false.

privilege*=*"string"*

-

ADD - EDIT - DELETE. The desired privilege to check on..

></IM:get.privileges.value >

Description:
Gets the user privilege for the current record. If the user has privilege displays its content, otherwise it doesn't display the contents.

---

get.recommendation.attribute

top

<IM:get.recommendation.attribute
attribute*=*"string"*

-

Name of attribute that should be displayed.

 Allowed Attribute values:
 recordid -- recordid of the recommendation
 casenumber -- case number
 completedby -- completed by user
 requestedby -- requested user
 title -- recommendation title
 text -- recommendation text
 comments -- comments if recommendation has been processed
 contentid -- content id if recommendation has been processed and a content record has been created
 docid - document id if recommendation has been processed and a content record has been created
 channel -- content channel assigned to recommendation
 status -- current status of this content recommendation
 priority --- current priority of this content recommendation
 dateadded -- creation date timestamp (java.util.Date)
 datemodified -- last modified timestamp (java.util.Date)

mask=*"string"*

-

Mask to use for date fields.

></IM:get.recommendation.attribute >

Description:
Tag that displays the selected attribute for a content recommendation. this tag must be included inside a get.recommendation.record tag.

---

get.recommendation.data

top

<IM:get.recommendation.data
dataset*=*"string"*
-
A unique identifier for the dataset. This identifier is used by the iterate.records tag to access records that are retrieved.

channels=*"string"*
-
List of channels separated by a + sign that the content recommendation query will search on.

documentid=*"string"*
-
Document ID the content recommendation query will search on.

completedby=*"string"*
-
If present, this tag will return only recommendations that where completed by this user. A valid value must be the user's login name.

requestedby=*"string"*
-
If present, this tag will return only recommendations that where requested by this user. A valid value must be the user's login name.

status=*"string"*
-
List of statuses separated b y a + sign. valid values are: new, duplicate, not_enough_info, unsuitable, other, created

priority=*"string"*
-
List of priorities separated b y a + sign. valid values are: low, medium, and high

sortby=*"string"*
-
List of sort parameters separated b y a + sign. Valid values are: dateadded, title, casenumber, priority, and status

direction="*string*"
-
The direction of the sort. Ascending or descending. The default if no value provided will be ascending.

casenumber="*string*"
-
returns content recommendations that have this casenumber

/>

Description:
Retrieves a list of content recommendations based on passed in parameters

---

get.recommendation.record

top

<IM:get.recommendation.record
recordid="*string*"
-
ID of record to retrieve.

id="*non-scriptable string*"
-
A desired prefix to use for all scripting variables created by this tag. There is no default id. Make sure all ids are unique

></IM:get.recommendation.record >

Description:
Tag that gets a recommendation record either from iteration or by a recommendation id.

---

get.record.masteridentifier

top

<IM:get.record.masteridentifier
id="*non-scriptable string*"

-
A desired prefix to use for all scripting variables created by this tag. The default value is 'masteridentifier'. If nesting tags, make sure all ids are unique

recordid=*"string"*
-
The content id for the text to be fetched. If it is a staging record prefix the recordid with S:

documentid=*"string"*
-
The document id of the content record.

localecode=*"string"*
-
If needed, the locale code for the text to be fetched.

></IM:get.record.masteridentifier >

Description:
Returns the master identifier text for a content record.

---

get.record.masteridentifier.value

top

<IM:get.record.masteridentifier.value
id=*"non-scriptable string"*
-
A desired prefix to use for all scripting variables created by this tag. The default value is 'masteridentifier'. If nesting tags, make sure all ids are unique

recordid=*"string"*
-
The content id for the text to be fetched. If it is a staging record prefix the recordid with S:

documentid=*"string"*
-
The document id of the content record.

localecode=*"string"*

-
If needed, the locale code for the text to be fetched.

></IM:get.record.masteridentifier.value >

Description:
Returns the master identifier text for a content record as a scripting variable.

---

get.repository.attribute

top

<IM:get.repository.attribute
id="*string*"
-
ID for this record.

attribute="*string*"
-
xpath of extended site attribute to return. Note that extended site properties are defined via schema in the admin tool.

systemattribute="*string*"
-
Basic site property value to return. Valid values are : <ol> repository - localized name of the repository refkey - reference key for site

view="*string*"
-
Reference key for the repository view to return property for

processbody="*true*" *or* "*false*"
-
set to 'true' if you create a body for this tag

></IM:get.repository.attribute >

Description:
Get a system or custom repository attribute. Only one of the attribute parameters can be specified at one time.

Example Usage:

```
<html>
<head>
   <meta content="<IM:get.repository.attribute attribute="demo/keywords" />" name="KEYWORDS">
   <title><IM:get.repository.attributes systemattribute="repository"/></title>
</head>
<body>
   ...
</body>
</html>
```

Scripting Variables:
none

Body Tags:
none

---

get.repository.attribute.value

top

<IM:get.repository.attribute.value
id=*"string"*
-
ID for this record.

attribute=*"string"*
-
xpath of extended site attribute to return. Note that extended site properties are defined via schema in the admin tool.

systemattribute=*"string"*
-
Basic site property value to return. Valid values are : <ol> domain - domain name of site refkey - locale neutral reference key for site

view=*"string"*
-
Reference key for the repository view to return property for

></IM:get.repository.attribute.value >

Description:
Get a system or custom domain attribute. Only one of the attribute parameters can be specified at one time.

Example Usage:

```
<html>
<head>
   <meta content="<IM:get.repository.attribute attribute="demo/keywords" />" name="KEYWORDS">
   <title><IM:get.repository.attributes systemattribute="domain"/></title>
</head>
<body>
   ...
</body>
</html>
```

Scripting Variables:
none

Body Tags:
none

---

get.resourcepath.content

top

<IM:get.resourcepath.content

></IM:get.resourcepath.content >

Description:
Prints to the file the resource path for the current content record.

---

get.resourcepath.user

top

<IM:get.resourcepath.user

></IM:get.resourcepath.user >

Description:
Prints to the file the resource path for the logged in user.

---

get.resourcepath.view

top

<IM:get.resourcepath.view
view*=*"string"*
-
The reference key of the desired view.

></IM:get.resourcepath.view >

Description:
Prints to the file the resource path for the specified view.

---

get.role.attribute

top

<IM:get.role.attribute
attribute*=*"string"*
-
Name of attribute that should be displayed.

 Allowed Attribute values:
 name -- returns the name of the role
 referencekey -- returns the reference key of the role
 recordid -- returns the recordid(guid) of the role

></IM:get.role.attribute >

Description:
Tag that displays the selected attribute for a role. this tag must be included inside a get.role.record tag.

---

get.role.data

top

<IM:get.role.data
dataset*=*"string"*
-
A unique identifier for the dataset, this identifier is used by the iterate.dataset tag to access records that are retrieved.

webroles=*"true" or "false"*
-
Set to true to returns all the web roles for this repository. The default behaviour is set to false.

userid=*"string"*
-
Record id of the user.

login=*"string"*
-
Login of the user.

/>

Description:
Retrieves a list of roles. If webroles is set to true it returns a list of all webroles for this repository, other wise set a user id or login to return a list of roles assigned to the desired user. If webroles is set to false and no user attribute is specified, this tag will attempt to get the security roles for the logged in user.

---

get.role.record

top

<IM:get.role.record
recordid=*"string"*
-
ID of record to retrieve.

role=*"string"*
-
Reference Key of Role to retrieve.

id=*"non-scriptable string"*
-
A desired prefix to use for all scripting variables created by this tag. There is no default id. If nesting role-record tags, make sure all ids are unique

></IM:get.role.record >

Description:
Tag that gets a role record from an iteration or from a passed in role identifier.

---

get.schema.data

top

<IM:get.schema.data
dataset*=*"string"*
-
A unique identifier for the dataset, this identifier is used by the iterator tag to access records that are retrieved.

type*=*"string"*
-
This is the schema type. Allowed values are CHANNEL(Regular channel schema), META(channel meta schema), USER, REPOSITORY

node=*"string"*
-
The parent node xpath. If not present this tag will assume that the parent node is the root.

referencekey=*"string"*
-
This value is necessary only if the schema is of type CHANNEL or META and it should be the reference key of the channel owner of the schema.

></IM:get.schema.data >

Description:

Retrieves a Schema Record with all its schema attributes. If a parent node is passed in it return the children of that node. If no node is passed in it assumes the root schema attribute.

---

get.schema.item

top

<IM:get.schema.item
id=*"non-scriptable string"*
-
A desired prefix to use for all scripting variables created by this tag. The default value is 'attributerecord'. If nesting tags, make sure all ids are unique

></IM:get.schema.item >

Description:
Tag that retrieves the current repetition item of type schema attribute

---

get.schema.item.attribute

top

<IM:get.schema.item.attribute
id=*"string"*
-
ID for this record.

attribute=*"string"*
-
Attribute that you are interested in. Allowed values are: XPATH, NAME, REFERENCEKEY, TYPE, TEXTHEIGHT, TEXTWIDTH

processbody=*"true" or "false"*
-
set to 'true' if you create a body for this tag

parentinlaw=
-

TODO

></IM:get.schema.item.attribute >

Description:
Returns the schema item attribute value for the passed in key

---

get.schema.item.attribute.value

top

<IM:get.schema.item.attribute.value
id="*string*"
-
ID for this record.

attribute="*string*"
-
Attribute that you are interested in. Allowed values are: XPATH, NAME, REFERENCEKEY, TYPE, TEXTHEIGHT, TEXTWIDTH

></IM:get.schema.item.attribute.value >

Description:
Returns the schema item attribute value for the passed in key

---

get.search.attribute

top

<IM:get.search.attribute
attribute="*string*"
-
Case sensitive xpath of iqxml to return.

id="*non-scriptable string*"
-

A desired prefix to use for all scripting variables created by this tag. The default value is 'searchattribute'. If nesting channel-attribute tags, make sure all ids are unique

processbody=*"true" or "false"*

-

set to 'true' if you create a body for this tag

></IM:get.search.attribute >

Description:
Displays properties inside the retuned IQXML. Either property or propertypath can be used...not both. Note that if no user is logged in, this tag will come back with no data.

```
<IM:userproperty property="firstname"/>
<IM:userproperty propertypath=xpath_to_address/>
```

---

get.search.data

top

<IM:get.search.data
channels=*"string"*

-

If used as a full text search this attribute is a delimited (+) list of Content Channels (using reference key) to include in the full text search. i.e. NEWS+EVENTS. If used as an attribute level search this attribute should only contain one channel reference key. i.e NEWS

categories=*"string"*

-

delimited (+) list of Content Categories (using reference key) to include in the full text search. i.e. CAT1+CAT2.

matchallcategories=*"true" or "false"*

-

Used in conjunction with 'categories' attribute. Set to true to 'and' the categories together in the search, and false to 'or' the categories.

matchallterms=*"true" or "false"*

-

FULL TEXT SEARCH: If this attribute is set to true, the search will AND all the words searched. If it is set to false, this tag will OR the words searched. The default value is true. ATTRIBUTE LEVEL SEARCH: If this

attribute is set to true, it will AND all of the search attributes. If it is set to false, this tag will OR the search attributes. The default value is true.

isattributelevel=*"true" or "false"*
-
Set to true if you would like to perform an attribute level search. Set to false to perform a full text search. The default value is false

views=*"string"*
-
delimited (+) list of Views (using reference key) to include in the full text search. i.e. VIEW1+VIEW2.

maxrecords=
-
Max amount of records returned by this search. Default is set to 200 records. A value of 0 returns all possible records

sortby=*"string"*
-
Sort By field for the results. Valid fields: contentchannel, docid, startdate, enddate, masteridentifier

direction=*"string"*
-
The direction of the sort. Ascending or descending. The default if no value provided will be ascending.

casenumber=*"string"*
-
Filter by the case link value that is assigned

></IM:get.search.data >

Description:
Tag that searches the repository for the passed in attribute. This tag performs a either a full text search or an attribute level search.

---

get.session.locale.attribute

top

<IM:get.session.locale.attribute

attribute=*"string"*
-
Locale attribute key. Allowed values are: code, value, description, dateformat, dateformatdisplay, timeformat, timeformatdisplay, and encoding.

 Attribute result Examples:

 Code = en_us
 Value = 1033
 description = English
 dateformat = %m/%d/%Y
 dateformatdisplay = mm/dd/yyyy
 timeformat = %I:%M %p
 timeformatdisplay = hh:mm
 encoding = UTF-8

></IM:get.session.locale.attribute >

Description:
Tag that displays the specified Locale attribute. if no attributes is specified it displays the Locale Code attribute(i.e. en_us)

---

get.session.locale.attribute.value

top

<IM:get.session.locale.attribute.value
id=*"non-scriptable string"*
-
A desired prefix to use for all scripting variables created by this tag. Make sure all ids are unique

></IM:get.session.locale.attribute.value >

Description:
Tag that creates scripting variables for the active locale. These are the scripting variables created: code, value, description, dateformat, dateformatdisplay, timeformat, timeformatdisplay, and encoding.

Variables Examples:

Code = en_us
Value = 1033
description = English
dateformat = %m/%d/%Y
dateformatdisplay = mm/dd/yyyy
timeformat = %I:%M %p
timeformatdisplay = hh:mm
encoding = UTF-8

---

get.shoppingcart.data

top

<IM:get.shoppingcart.data
cart*=*"string"*
-
Reference key for shopping cart.

dataset*=*"string"*
-
Dataset variable for the shopping cart

transactionid=*"string"*
-
Optional field for setting the transactionid if known. This attribute will get the transaction directly from the database instead of the user's session, or cookie.

></IM:get.shoppingcart.data >

Description:
Gets the shopping cart data

---

get.shoppingcart.option

top

<IM:get.shoppingcart.option
id=*"non-scriptable string"*
-
A desired prefix to use for all scripting variables created by this tag. There is no default id. If nesting option tags, make sure all ids are unique.

></IM:get.shoppingcart.option >

Description:
Gets the current record in the repetition of shopping cart options for record

---

get.shoppingcart.profile.attribute

top

<IM:get.shoppingcart.profile.attribute
attribute="string"
-
Case sensitive xpath of the cart schema attribute value.

></IM:get.shoppingcart.profile.attribute >

Description:
Tag that displays the desired cart schema attribute value for the specified transaction and xpath

---

get.shoppingcart.profile.attribute.value

top

<IM:get.shoppingcart.profile.attribute.value
attribute="string"
-
Case sensitive xpath of the cart schema attribute value.

id="non-scriptable string"
-
A desired prefix to use for all scripting variables created by this tag. The default value is 'attribute'. Make sure all ids are unique.

></IM:get.shoppingcart.profile.attribute.value >

Description:
Displays basic and extended user properties. Either property or propertypath can be used...not both. Note that if no user is logged in, this tag will come back with no data.

```
<IM:userproperty property="firstname"/>
<IM:userproperty propertypath=xpath_to_address/>
```

---

get.shoppingcart.profile.record

top

<IM:get.shoppingcart.profile.record
id="non-scriptable string"
-
A desired prefix to use for all scripting variables created by this tag. The default value is 'attribute'. Make sure all ids are unique.

cart*="string"
-
Reference key for shopping cart.

transactionid="string"
-
Optional field for setting the transactionid if known. This attribute will get the transaction directly from the database instead of the user's session, or cookie.

></IM:get.shoppingcart.profile.record >

Description:
Tag that gets the shopping cart profile document.

---

get.shoppingcart.record

top

<IM:get.shoppingcart.record
id="non-scriptable string"
-
A desired prefix to use for all scripting variables created by this tag. There is no default id. If nesting channel-item tags, make sure all ids are unique.

></IM:get.shoppingcart.record >

Description:
Gets the current record in the repetition of shopping cart records or if passed a recordid gets the content record itself

get.shoppingcart.record.exists

top

<IM:get.shoppingcart.record.exists
cart*="string"
-
Refrence key for shopping cart.

recordid*="string"
-
The record id of the channel record to be checked.

transactionid="string"
-
Optional field for setting the transactionid if known. This attribute will get the transaction directly from the database instead of the user's session, or cookie.

negate="non-scriptable string"
-
Reverses the conditional of this tag. If it evaluates to "true" it will show the contents of this tag if the value is not present. The default behaviour of this attribute is to be false.

></IM:get.shoppingcart.record.exists >

Description:
Checks if the channel record specified exists in the shopping cart.

get.shoppingcart.record.name

top

<IM:get.shoppingcart.record.name
type*="string"
-
Attribute that determines the type of field for this input HTML element. Allowed values for this attribute are: option, price, quantity, and schema. Set to <code>option</code> if field is meant to capture an option field for an item. Set to <code>price</code> if field is meant to capture the price field for an item. Set <code>quantity</

code> if field is meant to capture the quantity value for an item. Set to <code>schema</code> if the field is meant to capture information to be store in the predefine schema.

optionname=*"string"*
-
Attribute that is used to store the name of the desired option

attribute=*"string"*
-
If the type is set to be <code>schema</code>, set this value to be the xpath of the desired attribute in the schema

></IM:get.shoppingcart.record.name >

Description:
This tag returns the NAME value that options(html inputs) in the checkout screen should have.

---

get.static.resourcepath

top

<IM:get.static.resourcepath

></IM:get.static.resourcepath >

Description:
Prints to the specified static resource path for the active domain.

---

get.static.resourcepath.value

top

<IM:get.static.resourcepath.value
id=*"string"*
-
ID for this record.

></IM:get.static.resourcepath.value >

Description:
Gets the specified static resource path for the active domain.

---

get.subscription.name

top

<IM:get.subscription.name

></IM:get.subscription.name >

Description:
Tag that displays the name of the current subscription. This tag is intended to be a child of get.subscription.record tag

---

get.subscription.record

top

<IM:get.subscription.record

></IM:get.subscription.record >

Description:
Tag that gets the current subscription in the iteration.

---

get.user.attribute

top

<IM:get.user.attribute

attribute=*"string"*
-
Case sensitive xpath of extended user property to return. Note that extended user properties are defined via schema in the admin tool.

systemattribute=*"string"*
-
Basic user property value to return. Valid values IMe insensitive and are : <ol> firstname - users first name lastname - users last name email - users email address login - users login ID localecode - users preferred locale code (i.e. 1033) localedesc - users preferred locale description (i.e. en_US) defaultview - users default view

id=*"non-scriptable string"*
-
A desired prefix to use for all scripting variables created by this tag. The default value is 'attribute'. If nesting channel-attribute tags, make sure all ids are unique

processbody=*"true" or "false"*
-
set to 'true' if you create a body for this tag

></IM:get.user.attribute >

Description:
Displays basic and extended user properties. Either property or propertypath can be used...not both. Note that if no user is logged in, this tag will come back with no data.

```
<IM:userproperty property="firstname"/>
<IM:userproperty propertypath=xpath_to_address/>
```

---

get.user.attribute.value

top

<IM:get.user.attribute.value
attribute=*"string"*
-
Case sensitive xpath of extended user property to return. Note that extended user properties are defined via schema in the admin tool.

systemattribute=*"string"*

-
Basic user property value to return. Valid values IMe insensitive and are : <ol> firstname - users first name lastname - users last name email - users email address login - users login ID localecode - users preferred locale code (i.e. 1033) localedesc - users preferred locale description (i.e. en_US)

id=*"non-scriptable string"*
-
A desired prefix to use for all scripting variables created by this tag. The default value is 'attribute'. If nesting channel-attribute tags, make sure all ids are unique

></IM:get.user.attribute.value >

Description:
Displays basic and extended user properties. Either property or propertypath can be used...not both. Note that if no user is logged in, this tag will come back with no data.

```
<IM:userproperty property="firstname"/>
<IM:userproperty propertypath=xpath_to_address/>
```

---

get.user.data

top

<IM:get.user.data
dataset*=*"string"*
-
A unique identifier for the dataset, this identifier is used by the iterate.user.records tag to access records that are retrieved.

defaultviews=*"string"*
-
List of views delimited by a + sign that the ad-hoc query will search on. This means that will return user records that have assigned as their default views the passed in views.

where=*"string"*
-
Sql type where clause for fetching users. allowed values are firstName, lastName, login, and email. This are IMe sensitive key names and the fetch is also IMe senstive.
EXAMPLES: <IM:get.user.data dataset="userdataset" sortby="firstname" direction="ascending" where="login like 'g*' and lastName like 'R*'" />
 <IM:get.user.data dataset="userdataset" sortby=

"firstname" direction="ascending" where="firstName = 'John' and lastName like 'R*'" />

editorgroups=*"string"*
-
List of editorgroups delimited by a + sign that the query will search on.

sortby=*"string"*
-
This parameter determines which user record parameter will be used for sorting. Allowed values are: firstname, lastname, email, and login. The default behaviour is lastname

direction=*"string"*
-
The direction of the sort. Ascending or descending. The default if no value provided will be ascending.

categories=*"string"*
-
List of categories delimited by a + sign that the query will search on.

roles=*"string"*
-
List of roles delimited by a + sign that the query will search on.

localecode=*"string"*
-
Locale code for the current request. Locale code is of format en_US where en=language, US=location. Default is default locale of repository

maxrecords=
-
Max records to return.

/>

Description:
Retrieves a list of users using the DefaultViews and EditorGroups as query parameters.

---

get.user.has.role

top

<IM:get.user.has.role
roleid=*"string"*
-
The role id of the security role to be checked.

role=*"string"*
-
The role referencekey of the security role to be checked.

guid=*"string"*
-
The user id of the user to be checked.

userid=*"string"*
-
The login of the user to be checked.

negate=*"string"*
-
Reverses the conditional of this tag. If it evaluates to "true" it will show the contents of this tag if the value is not present. The default behaviour of this attribute is to be false.

></IM:get.user.has.role >

Description:
Checks if the specified user has the specified role. You can specify the user by userid, login. If you don't specify a user attribute, and this tag is inside a get.user.record, it will be applied to the user object of the get.user.record. If it is not inside a get.user.record, it will be applied to the logged in user. You can specify the role by roleid, role reference key, or if this tag is inside a get.role.record, it will be applied to the role object of the get.role.record.

---

get.user.keyvalue

top

<IM:get.user.keyvalue
id*=*"non-scriptable string"*
-

A desired prefix to use for all scripting variables created by this tag. There is no default id. Make sure all ids are unique

recordid=*"string"*
-
The userid for the user we are getting the key value pair for

login=*"string"*
-
The login id for the user we are getting the key value pair for

/>

Description:
Tag that gets a key/value pair

---

get.user.record

top

<IM:get.user.record
recordid=*"string"*
-
ID of record to retrieve.

login=*"string"*
-
Login name of user record to retrieve.

generatexml=*"true" or "false"*
-
Set to true to generate the xml used in get.user.attribute tag. If get.user.attribute is not being used, setting this parameter to false will increase performance. Default is true.

id=*"non-scriptable string"*
-
A desired prefix to use for all scripting variables created by this tag. There is no default id. If nesting user-record tags, make sure all ids are unique

parentinlaw=

-
TODO

></IM:get.user.record >

Description:
Tag that gets a user record either from a iterate.user.records iteration, from a passed recordid, or a passed in login name.

---

get.view.template

top

<IM:get.view.template
view*=*"string"*
-
The reference key of the desired view.

></IM:get.view.template >

Description:
Returns the template for the passed in view, if no view is passed in it returns the template for the domain.

---

get.view.template.value

top

<IM:get.view.template.value
view*=*"string"*
-
The reference key of the desired view.

id*=*"string"*
-
ID for this record.

></IM:get.view.template.value >

Description:
Returns the template for the passed in view, if no view is passed in it returns the template for the domain.

---

get.views

top

<IM:get.views
dataset*=*"string"*
-
Name of the dataset that contains the department list information

startview=*"string"*
-
Referencekey of the site to start on

/>

Description:
Get a list of all compartments for the current domain.

Example Usage:
<IM:get.departments dataset="departments"/>

Scripting Variables:
none

Body Tags:
none

---

get.wizard.previous.response

top

<IM:get.wizard.previous.response
id*=*"string"*
-
The unique id representing the PreviousResponse

></IM:get.wizard.previous.response >

Description:
Tag that receives the current PreviousResponse object from inside the iterate.wizard.previous.responses tag.

---

get.wizardfield.record

top

<IM:get.wizardfield.record
id*=*"string"*
-
The unique id representing the WizardFieldObject

></IM:get.wizardfield.record >

Description:
Tag that receives the current WizardFieldObject object from inside the iterate.wizardform.fields tag.

---

has.channel.category

top

<IM:has.channel.category
channel=*"string"*
-
Reference key of the Content Channel

channelid=*"string"*
-
Record id of the Content Channel

category=*"string"*
-
Reference key of the category to test

categoryid=*"string"*

-
Record id of the category to test.

negate=*"string"*
-
Reverses the conditional of this tag. If it evaluates to "true" it will show the contents of this tag if the value is not present. The default behaviour of this attribute is to be false.

id=*"non-scriptable string"*
-
A desired prefix to use for all scripting variables created by this tag. There is no default id. Make sure all ids are unique

></IM:has.channel.category >

Description:
Tag that tests if the current category is part of the categories of a content channel.

---

index.next

top

<IM:index.next
ifnull=*"true" or "false"*
-

></IM:index.next >

Description:
Create HTML hyperlink to the next batch of records from a iterate.channel.records tag.

Example Usage:

```
<table width="100%" cellpadding="0" cellspacing="0">
   <IM:iterate.channel.records dataset="news" maxpageitems="10" maxindexpages="20">
      <IM:get.channel.record id="newsrec">
         <tr>
            <td align="left">
               <a href="index?page=detail&rec=<%=newsrec.recordid%>"><IM:get.channel.attribute attribute=
"news/title"/></a>
            </td>
         </tr>
```

```
      </IM:get.channel.record>
      <IM:iterate.index>
         <tr>
            <td>
               <br>
               Result Pages:
               <IM:index.prev>
                      <a href="<%= pageUrl %>">[<< Prev]</a>
               </IM:index.prev>
               <IM:index.pages>
                  <% if (pageNumber.intValue() < 10) { %> <% } %>
                  <% if (pageNumber == iteratorpagenumber) { %>
                     <b><%= pageNumber %></b>
                  <% } else { %>
                        <a href="<%= pageUrl %>"><%= pageNumber %></a>
                  <%}%>
               </IM:index.pages>
               <IM:index.next>
                   <a href="<%= pageUrl %>">[Next >>]</a>
               </IM:index.next>
               <br>
            </td>
         </tr>
      </IM:iterate.index>
   </IM:iterate.channel.records>
</table>
```

Scripting Variables:
none

Body Tags:
none

---

index.pages

top

<IM:index.pages

></IM:index.pages >

Description:
Create HTML hyperlink to each batch page of records from a iterate.channel.records tag.

Example Usage:

```
<table width="100%" cellpadding="0" cellspacing="0">
   <IM:iterate.channel.records dataset="news" maxpageitems="10" maxindexpages="20">
      <IM:get.channel.record id="newsrec">
         <tr>
            <td align="left">
               <a href="index?page=detail&rec=<%=newsrec.recordid%>"><IM:get.channel.attribute attribute=
"news/title"/></a>
            </td>
         </tr>
      </IM:get.channel.record>
      <IM:iterate.index>
         <tr>
            <td>
               <br>
               Result Pages:
               <IM:index.prev>
                      <a href="<%= pageUrl %>">[<< Prev]</a>
               </IM:index.prev>
               <IM:index.pages>
                  <% if (pageNumber.intValue() < 10) { %> <% } %>
                  <% if (pageNumber == iteratorpagenumber) { %>
                     <b><%= pageNumber %></b>
                  <% } else { %>
                        <a href="<%= pageUrl %>"><%= pageNumber %></a>
                  <%}%>
               </IM:index.pages>
               <IM:index.next>
                   <a href="<%= pageUrl %>">[Next >>]</a>
               </IM:index.next>
               <br>
            </td>
         </tr>
      </IM:iterate.index>
   </IM:iterate.channel.records>
</table>
```

Scripting Variables:
none

Body Tags:
none

index.prev

top

<IM:index.prev
ifnull=*"true" or "false"*
-

></IM:index.prev >

Description:
Create HTML hyperlink to page with previous batch of records from a iterate.channel.records tag.

Example Usage:

```
<table width="100%" cellpadding="0" cellspacing="0">
   <IM:iterate.channel.records dataset="news" maxpageitems="10" maxindexpages="20">
      <IM:get.channel.record id="newsrec">
         <tr>
            <td align="left">
               <a href="index?page=detail&rec=<%=newsrec.recordid%>"><IM:get.channel.attribute attribute=
"news/title"/></a>
            </td>
         </tr>
      </IM:get.channel.record>
      <IM:iterate.index>
         <tr>
            <td>
               <br>
               Result Pages:
               <IM:index.prev>
                      <a href="<%= pageUrl %>">[<< Prev]</a>
               </IM:index.prev>
               <IM:index.pages>
                  <% if (pageNumber.intValue() < 10) { %> <% } %>
                  <% if (pageNumber == iteratorpagenumber) { %>
                     <b><%= pageNumber %></b>
                  <% } else { %>
                        <a href="<%= pageUrl %>"><%= pageNumber %></a>
                  <%}%>
               </IM:index.pages>
               <IM:index.next>
                   <a href="<%= pageUrl %>">[Next >>]</a>
               </IM:index.next>
               <br>
            </td>
         </tr>
```

```
      </IM:iterate.index>
   </IM:iterate.channel.records>
</table>
```

Scripting Variables:
none

Body Tags:
none

---

input.channel.contribution

top

<IM:input.channel.contribution
attribute*=*"string"*
-
Path to custom user attribute. To contribute display start and end dates, or event start and end dates, you must specify one of the following keywords and a date mask DISPLAYSTARTDATE, DISPLAYENDDATE, EVENTSTARTDATE, EVENTENDDATE.

mask=*"string"*
-
Date mask for Content Dates.

size=*"string"*
-
Size of the input field.

maxlength=*"string"*
-
Maximum length of the input field.

value=*"string"*
-
Initial value of the input field. Note : if nothing is entered for the value, the value will get prefilled by the information in the database.

css=*"string"*
-
CSS Class to be used for the input field

cols=*"string"*
-
If this input is a text area, this attribute defines how many columns it will have

rows=*"string"*
-
If this input is a text area, this attribute defines how many rows it will have

/>

Description:
Creates an HTML input element for an attribute of an existing content channel.

Example Usage:
<IM:form.channel.contribution channel="news" departments="department1+department2" success="thankyou" error="errorpage">
Title:
<IM:input.channel.contribution attribute="/news/title"/>
Body
<IM:input.channel.contribution attribute="/news/body"/>
</IM:channel.contribute.form>

Scripting Variables:
none

Body Tags:
input.channel.contribution

---

input.dataform.answer

top

<IM:input.dataform.answer
css=*"string"*
-
CSS Class to be used for the input field

inputtext=*"string"*
-
Value to be displayed in the field

></IM:input.dataform.answer >

Description:

Create HTML input element that matches answer type with predefined answers.

Example Usage:
Displaying the Form

```
<IM:form.dataform dataform="INQUIRIES" success="requestthank" error="requestthank">
   <IM:iterate.dataform.question>
      <IM:get.dataform.question.record>
         <b><IM:get.dataform.question/></b><br>
            <IM:iterate.dataform.answer>
               <IM:get.dataform.answer.record>
                  <IM:input.dataform.answer/><br>
               </IM:get.dataform.answer.record>
            </IM:iterate.dataform.answer>
      </IM:get.dataform.question.record>
   </IM:iterate.dataform.question> <br>
<input type="submit" value="Submit">
</IM:form.dataform>
```

Displaying Results

```
<IM:get.dataform.results name="nameofform" individual="false" aggregate="false">
   <IM:iterate.dataform.question>
      <IM:get.dataform.question.record>
         <b><IM:get.dataform.question/></b><br>
            <IM:iterate.dataform.answer>
               <IM:get.dataform.answer.record>
                  <IM:get.dataform.answer/><br>
               </IM:get.dataform.answer.record>
            </IM:iterate.dataform.answer>
      </IM:get.dataform.question.record>
   </IM:iterate.dataform.question>
</IM:get.dataform.results>
```

Scripting Variables:
none

Body Tags:
none

input.emailsubscription

top

<IM:input.emailsubscription
css=*"string"*
-
CSS Class to be used for the input field

showtext=*"true" or "false"*
-
If evaluates to true, the text of the current subscription will be displayed next to the checkbox. Otherwise only the checkbox will be displayed. In this IMe please use the get.subscription.name tag to get the subscription name. If this argument is omitted, the default value is set to true.

checked=*"true" or "false"*
-
If evaluates to true, all the checkboxes in the repetition will be pre-checked.

></IM:input.emailsubscription >

Description:
Tag that display a checkbox for the current subscription in the repetition

---

input.message.author

top

<IM:input.message.author
size=*"string"*
-
Size of the input field.

maxlength=*"string"*
-
Maximum length of the input field.

value*=*"string"*
-
Initial value of the input field.

/>

Description:
Creates an HTML INPUT element for a message author.

Example Usage:
Displaying All Messages**:**

```
 <!-- get message data  -->
<IM:get.message.data dataset="data" topic="hammers"/>
<!-- iterate through messages in discussion records  -->
<IM:iterate.message.records dataset="data">
   <!-- get handle to individual record -->
   <IM:get.message.record>
      <!-- space message to show hierarchy and display message title and text-->
      <img src="images\spacer.gif" width="<% =message.depth %>" aligh="left"/><IM:get.message.title/> <br>
      <img src="images\spacer.gif" width="<% =message.depth %>" aligh="left"/><IM:get.message.text/> <br>
      <!-- display a reply link that passes the message record id to the reply page -->
      <a href="index?page=messagereply&rec=<%=message.recordid%>">Reply</a> <br>
   </IM:get.message.record>
</IM:iterate.message.records>
```

Replying to a Message:

```
<!-- declare page variable and load recordid parameter off url from previous page -->
<% String rec = request.getParameter("rec");%>
<!-- create HTML form for input fields -->
<IM:form.message parentid="<%=rec%>" success="successpage" error="errorpage">
   <!-- display input fields for title and text -->
   Title<br>
   <input.message.title><br>
   Text<br>
   <input.message.text><br>
   <!-- display submit button -->
   <input type="submit" value="Submit">
</IM:form.message>
```

Scripting Variables:
none

Body Tags:
none

---

input.message.text

top

<IM:input.message.text
size=*"string"*
-
Size of the input field.

maxlength=*"string"*
-
Maximum length of the input field.

value*=*"string"*
-
Initial value of the input field.

/>

Description:
Creates an HTML INPUT element for a message body text.

Example Usage:
Displaying All Messages**:**

```
 <!-- get message data  -->
<IM:get.message.data dataset="data" topic="hammers"/>
<!-- iterate through messages in discussion records  -->
<IM:iterate.message.records dataset="data">
   <!-- get handle to individual record -->
   <IM:get.message.record>
      <!-- space message to show hierarchy and display message title and text-->
      <img src="images\spacer.gif" width="<% =message.depth %>" aligh="left"/><IM:get.message.title/> <br>
      <img src="images\spacer.gif" width="<% =message.depth %>" aligh="left"/><IM:get.message.text/> <br>
      <!-- display a reply link that passes the message record id to the reply page -->
      <a href="index?page=messagereply&rec=<%=message.recordid%>">Reply</a> <br>
   </IM:get.message.record>
</IM:iterate.message.records>
```

Replying to a Message:

```
<!-- declare page variable and load recordid parameter off url from previous page -->
<% String rec = request.getParameter("rec");%>
<!-- create HTML form for input fields -->
<IM:form.message parentid="<%=rec%>" success="successpage" error="errorpage">
   <!-- display input fields for title and text -->
   Title<br>
   <input.message.title><br>
   Text<br>
   <input.message.text><br>
   <!-- display submit button -->
```

```
   <input type="submit" value="Submit">
</IM:form.message>
```

Scripting Variables:
none

Body Tags:
none

---

input.message.title

top

<IM:input.message.title
size=*"string"*
-
Size of the input field.

maxlength=*"string"*
-
Maximum length of the input field.

value*=*"string"*
-
Initial value of the input field.

/>

Description:
Creates an HTML INPUT element for a message title.

Example Usage:
Displaying All Messages**:**

```
 <!-- get message data  -->
<IM:get.message.data dataset="data" topic="hammers"/>
<!-- iterate through messages in discussion records  -->
<IM:iterate.message.records dataset="data">
   <!-- get handle to individual record -->
   <IM:get.message.record>
      <!-- space message to show hierarchy and display message title and text-->
      <img src="images\spacer.gif" width="<% =message.depth %>" aligh="left"/><IM:get.message.title/> <br>
      <img src="images\spacer.gif" width="<% =message.depth %>" aligh="left"/><IM:get.message.text/> <br>
      <!-- display a reply link that passes the message record id to the reply page -->
```

```
      <a href="index?page=messagereply&rec=<%=message.recordid%>">Reply</a> <br>
   </IM:get.message.record>
</IM:iterate.message.records>
```

Replying to a Message:

```
<!-- declare page variable and load recordid parameter off url from previous page -->
<% String rec = request.getParameter("rec");%>
<!-- create HTML form for input fields -->
<IM:form.message parentid="<%=rec%>" success="successpage" error="errorpage">
   <!-- display input fields for title and text -->
   Title<br>
   <input.message.title><br>
   Text<br>
   <input.message.text><br>
   <!-- display submit button -->
   <input type="submit" value="Submit">
</IM:form.message>
```

Scripting Variables:
none

Body Tags:
none

---

input.recommendation.contribution

top

<IM:input.recommendation.contribution
attribute*=*"string"*
-
Attribute to contribute. Valid values are: title -- Recommendation Title
 text -- Recommendation text casenumber --Case Num channel -- Content Channel for recommendation categories -- Categories associated with recommendation priority -- priority for recommendation locale -- locale code for recommendation

size=*"string"*
-
Size of the input field.

maxlength=*"string"*
-
Maximum length of the input field.

value=*"string"*

-
Initial value of the input field. Note : if nothing is entered for the value, the value will get prefilled by the information in the database.

css=*"string"*
-
CSS Class to be used for the input field

cols=*"string"*
-
If this input is a text area, this attribute defines how many columns it will have

rows=*"string"*
-
If this input is a text area, this attribute defines how many rows it will have

/>

Description:
Creates an HTML input element for a content recommendation attribute

---

input.search.attribute

top

<IM:input.search.attribute
attribute*=*"string"*
-

/>

Description:
Creates an HTML INPUT submit element for performing attribute searches on a content channel.

Example Usage:
Capturing Search Text
<IM:form.search.attribute channel="directory" matchtype="ALL" success="searchresults" error="errorpage">
First Name:
<IM:input.search.attribute attribute="directory\first_name"/>


```
   Last Name:<br>
   <IM:input.search.attribute attribute="directory\last_name"/>
   <br>
   <input type="submit"/>
</IM:form.search.attribute>
```

Displaying Search Results:

```
<IM:iterate.channel.records dataset="searchresults" maxpageitems="2">
<IM:get.channel.record id="currentrecord">
   <b>Channel: <IM:get.channel.attribute attribute="TYPE" /></b><br>
   <a href="index?page=detail&guid=<%=currentrecord.recordid%>"><IM:get.channel.attribute attribute=
"TITLE" /></a><br>
   </IM:get.channel.record>
</IM:iterate.channel.records>
```

Scripting Variables:
none

Body Tags:
none

---

input.search.fulltext

top

```
<IM:input.search.fulltext
size="string"
-


maxlength="string"
-


value="string"
-


/>
```

Description:
Creates an HTML INPUT element for capturing the full text search text

Example Usage:
Capturing Search Text:

```
<IM:form.search.fulltext channels="news+events+press" success="searchresults" error="errorpage">
   Search String: <br>
   <IM:input.search.fulltext/>
   <br>
   <input type="submit"/>
</IM:form.search.fulltext>
```

Displaying Search Results:

```
<IM:iterate.channel.records dataset="searchresults" maxpageitems="2">
<IM:get.channel.record id="currentrecord">
   <b>Channel: <IM:get.channel.attribute attribute="TYPE" /></b><br>
   <a href="index?page=detail&guid=<%=currentrecord.recordid%>"><IM:get.channel.attribute attribute=
"TITLE" /></a><br>
   </IM:get.channel.record>
</IM:iterate.channel.records>
```

Scripting Variables:
none

Body Tags:
none

---

input.subscription

top

<IM:input.subscription
css=*"string"*
-
CSS Class to be used for the input field

showtext=*"true" or "false"*
-
If evaluates to true, the text of the current subscription will be displayed next to the checkbox. Otherwise only the checkbox will be displayed. In this IMe please use the get.subscription.name tag to get the subscription name. If this argument is omitted, the default value is set to true.

checked=*"true" or "false"*
-
If evaluates to true, all the checkboxes in the repetition will be pre-checked.

></IM:input.subscription >

Description:
Tag that display a checkbox for the current subscription in the repetition

---

input.subscription.email

top

<IM:input.subscription.email
size=*"string"*
-
Size of the input field.

maxlength=*"string"*
-
Maximum length of the input field.

css=*"string"*
-
CSS Class to be used for the input field

/>

Description:
Creates an HTML input element for an email address to be search for. Typically used for searching email address in the subscribe/unsubscribe process

---

input.user.attribute

top

<IM:input.user.attribute
attribute*=*"string"*
-
Path to custom user attribute.

size=*"string"*

-
Size of the input field.

maxlength=*"string"*
-
Maximum length of the input field.

value*=*"string"*
-
Initial value of the input field. Note : if nothing is entered for the value, the value will get prefilled by the information in the database.

css=*"string"*
-
CSS Class to be used for the input field

cols=*"string"*
-
If this input is a text area, this attribute defines how many columns it will have

rows=*"string"*
-
If this input is a text area, this attribute defines how many rows it will have

/>

Description:
Creates an HTML input element for a custom user attribute.

Example Usage:

```
<IM:form.user.attributes  success="successpage" error="errorpage">
   User ID: <IM:input.user.id.error/><br>
   <IM:input.user.id value="" system size="20"/><br>
   Password: <IM:input.user.password .error/><br>
   <IM:input.user.password value="" size="20"/><br>
   First Name: <IM:input.user.firstname .error/><br>
   <IM:input.firstname value="" size="30"/><br>
   Last Name: <IM:input.user.lastname .error/><br>
   <IM:input.user.lastname value="" size="30"/><br>
   Email Address: <IM:input.user.email.error/><br>
   <IM:input.user.email value="" size="30"/><br>
   Phone Number:<br>
   <IM:input.user.attribute attribute="demo/phone" value="" size="10"/><br>
   Zip Code<br>
   <IM:input.user.attribute attribute="demo/zipcode" value="" size="5"/><br>
   <br>
```

```
   <IM:input.submit value="Save Changes"/>
</IM:form.user.attributes>
```

Scripting Variables:
none

Body Tags:
none

---

input.user.email

top

<IM:input.user.email
size=*"string"*
-
Size of the input field.

maxlength=*"string"*
-
Maximum length of the input field.

value*=*"string"*
-
Initial value of the input field. Note : if nothing is entered for the value, the value will get prefilled by the information in the database.

css=*"string"*
-
CSS Class to be used for the input field

/>

Description:
Creates an HTML input element for a users email address.

Example Usage:

```
<IM:form.user.attributes  success="successpage" error="errorpage">
   User ID: <IM:input.user.id.error/><br>
   <IM:input.user.id value="" system size="20"/><br>
   Password: <IM:input.user.password .error/><br>
   <IM:input.user.password value="" size="20"/><br>
```

```
   First Name: <IM:input.user.firstname .error/><br>
   <IM:input.firstname value="" size="30"/><br>
   Last Name: <IM:input.user.lastname .error/><br>
   <IM:input.user.lastname value="" size="30"/><br>
   Email Address: <IM:input.user.email.error/><br>
   <IM:input.user.email value="" size="30"/><br>
   Phone Number:<br>
   <IM:input.user.attribute attribute="demo/phone" value="" size="10"/><br>
   Zip Code<br>
   <IM:input.user.attribute attribute="demo/zipcode" value="" size="5"/><br>
   <br>
   <IM:input.submit value="Save Changes"/>
</IM:form.user.attributes>
```

Scripting Variables:
none

Body Tags:
none

---

input.user.email.error

top

<IM:input.user.email.error

/>

Description:
Displays email errors message from Action requests when user enters incorrect email address.

Example Usage:

```
<IM:form.user.attributes  success="successpage" error="errorpage">
   User ID: <IM:input.user.id.error/><br>
   <IM:input.user.id value="" system size="20"/><br>
   Password: <IM:input.user.password .error/><br>
   <IM:input.user.password value="" size="20"/><br>
   First Name: <IM:input.user.firstname .error/><br>
   <IM:input.firstname value="" size="30"/><br>
   Last Name: <IM:input.user.lastname .error/><br>
   <IM:input.user.lastname value="" size="30"/><br>
   Email Address: <IM:input.user.email.error/><br>
   <IM:input.user.email value="" size="30"/><br>
```

```
    Phone Number:<br>
    <IM:input.user.attribute attribute="demo/phone" value="" size="10"/><br>
    Zip Code<br>
    <IM:input.user.attribute attribute="demo/zipcode" value="" size="5"/><br>
    <br>
    <IM:input.submit value="Save Changes"/>
</IM:form.user.attributes>
```

Scripting Variables:
none

Body Tags:
none

---

input.user.firstname

top

<IM:input.user.firstname
size=*"string"*
-
Size of the input field.

maxlength=*"string"*
-
Maximum length of the input field.

value*=*"string"*
-
Initial value of the input field. Note : if nothing is entered for the value, the value will get prefilled by the information in the database.

css=*"string"*
-
CSS Class to be used for the input field

/>

Description:
Creates an HTML input element for a users fist name.

Example Usage:

```
<IM:form.user.attributes  success="successpage" error="errorpage">
   User ID: <IM:input.user.id.error/><br>
   <IM:input.user.id value="" system size="20"/><br>
   Password: <IM:input.user.password .error/><br>
   <IM:input.user.password value="" size="20"/><br>
   First Name: <IM:input.user.firstname .error/><br>
   <IM:input.firstname value="" size="30"/><br>
   Last Name: <IM:input.user.lastname .error/><br>
   <IM:input.user.lastname value="" size="30"/><br>
   Email Address: <IM:input.user.email.error/><br>
   <IM:input.user.email value="" size="30"/><br>
   Phone Number:<br>
   <IM:input.user.attribute attribute="demo/phone" value="" size="10"/><br>
   Zip Code<br>
   <IM:input.user.attribute attribute="demo/zipcode" value="" size="5"/><br>
   <br>
   <IM:input.submit value="Save Changes"/>
</IM:form.user.attributes>
```

Scripting Variables:
none

Body Tags:
none

---

input.user.firstname.error

top

```
<IM:input.user.firstname.error


/>
```

Description:
Displays errors message from action requests when no fist name is entered.

Example Usage:

```
<IM:form.user.attributes  success="successpage" error="errorpage">
   User ID: <IM:input.user.id.error/><br>
   <IM:input.user.id value="" system size="20"/><br>
   Password: <IM:input.user.password .error/><br>
   <IM:input.user.password value="" size="20"/><br>
```

```
   First Name: <IM:input.user.firstname .error/><br>
   <IM:input.firstname value="" size="30"/><br>
   Last Name: <IM:input.user.lastname .error/><br>
   <IM:input.user.lastname value="" size="30"/><br>
   Email Address: <IM:input.user.email.error/><br>
   <IM:input.user.email value="" size="30"/><br>
   Phone Number:<br>
   <IM:input.user.attribute attribute="demo/phone" value="" size="10"/><br>
   Zip Code<br>
   <IM:input.user.attribute attribute="demo/zipcode" value="" size="5"/><br>
   <br>
   <IM:input.submit value="Save Changes"/>
</IM:form.user.attributes>
```

Scripting Variables:
none

Body Tags:
none

---

input.user.id

top

<IM:input.user.id
size=*"string"*
-
Size of the input field.

maxlength=*"string"*
-
Maximum length of the input field.

value*=*"string"*
-
Initial value of the input field. Note : if nothing is entered for the value, the value will get prefilled by the information in the database.

css=*"string"*
-
CSS Class to be used for the input field

/>

Description:
Creates an HTML input element for a user id.

Example Usage:

```
<IM:form.user.attributes  success="successpage" error="errorpage">
   User ID: <IM:input.user.id.error/><br>
   <IM:input.user.id value="" system size="20"/><br>
   Password: <IM:input.user.password .error/><br>
   <IM:input.user.password value="" size="20"/><br>
   First Name: <IM:input.user.firstname .error/><br>
   <IM:input.firstname value="" size="30"/><br>
   Last Name: <IM:input.user.lastname .error/><br>
   <IM:input.user.lastname value="" size="30"/><br>
   Email Address: <IM:input.user.email.error/><br>
   <IM:input.user.email value="" size="30"/><br>
   Phone Number:<br>
   <IM:input.user.attribute attribute="demo/phone" value="" size="10"/><br>
   Zip Code<br>
   <IM:input.user.attribute attribute="demo/zipcode" value="" size="5"/><br>
   <br>
   <IM:input.submit value="Save Changes"/>
</IM:form.user.attributes>
```

Scripting Variables:
none

Body Tags:
none

---

input.user.id.error

top

<IM:input.user.id.error

/>

Description:
Displays errors message from action requests when no user id is entered.

Example Usage:

```
<IM:form.user.attributes  success="successpage" error="errorpage">
   User ID: <IM:input.user.id.error/><br>
   <IM:input.user.id value="" system size="20"/><br>
   Password: <IM:input.user.password .error/><br>
   <IM:input.user.password value="" size="20"/><br>
   First Name: <IM:input.user.firstname .error/><br>
   <IM:input.firstname value="" size="30"/><br>
   Last Name: <IM:input.user.lastname .error/><br>
   <IM:input.user.lastname value="" size="30"/><br>
   Email Address: <IM:input.user.email.error/><br>
   <IM:input.user.email value="" size="30"/><br>
   Phone Number:<br>
   <IM:input.user.attribute attribute="demo/phone" value="" size="10"/><br>
   Zip Code<br>
   <IM:input.user.attribute attribute="demo/zipcode" value="" size="5"/><br>
   <br>
   <IM:input.submit value="Save Changes"/>
</IM:form.user.attributes>
```

Scripting Variables:
none

Body Tags:
none

---

input.user.lastname

top

<IM:input.user.lastname
size=*"string"*
-
Size of the input field.

maxlength=*"string"*
-
Maximum length of the input field.

value*=*"string"*
-

Initial value of the input field. Note : if nothing is entered for the value, the value will get prefilled by the information in the database.

css=*"string"*
-
CSS Class to be used for the input field

/>

Description:
Creates an HTML input element for a user last name.

Example Usage:

```
<IM:form.user.attributes  success="successpage" error="errorpage">
   User ID: <IM:input.user.id.error/><br>
   <IM:input.user.id value="" system size="20"/><br>
   Password: <IM:input.user.password .error/><br>
   <IM:input.user.password value="" size="20"/><br>
   First Name: <IM:input.user.firstname .error/><br>
   <IM:input.firstname value="" size="30"/><br>
   Last Name: <IM:input.user.lastname .error/><br>
   <IM:input.user.lastname value="" size="30"/><br>
   Email Address: <IM:input.user.email.error/><br>
   <IM:input.user.email value="" size="30"/><br>
   Phone Number:<br>
   <IM:input.user.attribute attribute="demo/phone" value="" size="10"/><br>
   Zip Code<br>
   <IM:input.user.attribute attribute="demo/zipcode" value="" size="5"/><br>
   <br>
   <IM:input.submit value="Save Changes"/>
</IM:form.user.attributes>
```

Scripting Variables:
none

Body Tags:
none

---

input.user.lastname.error

top

<IM:input.user.lastname.error

/>

Description:
Displays errors message from action requests when no last name is entered.

Example Usage:

```
<IM:form.user.attributes  success="successpage" error="errorpage">
   User ID: <IM:input.user.id.error/><br>
   <IM:input.user.id value="" system size="20"/><br>
   Password: <IM:input.user.password .error/><br>
   <IM:input.user.password value="" size="20"/><br>
   First Name: <IM:input.user.firstname .error/><br>
   <IM:input.firstname value="" size="30"/><br>
   Last Name: <IM:input.user.lastname .error/><br>
   <IM:input.user.lastname value="" size="30"/><br>
   Email Address: <IM:input.user.email.error/><br>
   <IM:input.user.email value="" size="30"/><br>
   Phone Number:<br>
   <IM:input.user.attribute attribute="demo/phone" value="" size="10"/><br>
   Zip Code<br>
   <IM:input.user.attribute attribute="demo/zipcode" value="" size="5"/><br>
   <br>
   <IM:input.submit value="Save Changes"/>
</IM:form.user.attributes>
```

Scripting Variables:
none

Body Tags:
none

---

input.user.password

top

<IM:input.user.password
size=*"string"*
-
Size of the input field.

maxlength=*"string"*

-
Maximum length of the input field.

value*=*"string"*
-
Initial value of the input field. Note : the password will NEVER be prefilled by the database

css=*"string"*
-
CSS Class to be used for the input field

/>

Description:
Creates an HTML input element for a user password.

Example Usage:

```
<IM:form.user.attributes  success="successpage" error="errorpage">
   User ID: <IM:input.user.id.error/><br>
   <IM:input.user.id value="" system size="20"/><br>
   Password: <IM:input.user.password .error/><br>
   <IM:input.user.password value="" size="20"/><br>
   First Name: <IM:input.user.firstname .error/><br>
   <IM:input.firstname value="" size="30"/><br>
   Last Name: <IM:input.user.lastname .error/><br>
   <IM:input.user.lastname value="" size="30"/><br>
   Email Address: <IM:input.user.email.error/><br>
   <IM:input.user.email value="" size="30"/><br>
   Phone Number:<br>
   <IM:input.user.attribute attribute="demo/phone" value="" size="10"/><br>
   Zip Code<br>
   <IM:input.user.attribute attribute="demo/zipcode" value="" size="5"/><br>
   <br>
   <IM:input.submit value="Save Changes"/>
</IM:form.user.attributes>
```

Scripting Variables:
none

Body Tags:
none

input.user.password.error

top

<IM:input.user.password.error

/>

Description:
Displays errors message from action requests when no password is entered.

Example Usage:

```
<IM:form.user.attributes  success="successpage" error="errorpage">
   User ID: <IM:input.user.id.error/><br>
   <IM:input.user.id value="" system size="20"/><br>
   Password: <IM:input.user.password .error/><br>
   <IM:input.user.password value="" size="20"/><br>
   First Name: <IM:input.user.firstname .error/><br>
   <IM:input.firstname value="" size="30"/><br>
   Last Name: <IM:input.user.lastname .error/><br>
   <IM:input.user.lastname value="" size="30"/><br>
   Email Address: <IM:input.user.email.error/><br>
   <IM:input.user.email value="" size="30"/><br>
   Phone Number:<br>
   <IM:input.user.attribute attribute="demo/phone" value="" size="10"/><br>
   Zip Code<br>
   <IM:input.user.attribute attribute="demo/zipcode" value="" size="5"/><br>
   <br>
   <IM:input.submit value="Save Changes"/>
</IM:form.user.attributes>
```

Scripting Variables:
none

Body Tags:
none

---

input.user.value

top

<IM:input.user.value
size=*"string"*

-
Size of the input field.

maxlength=*"string"*
-
Maximum length of the input field.

value*=*"string"*
-
Initial value of the input field. Note : if nothing is entered for the value, the value will get prefilled by the information in the database.

css=*"string"*
-
CSS Class to be used for the input field

attribute=*"string"*
-
Attribute to be entered. Allowed values firstname, lastname, login, email, password, showname, showemail, alias, avatar, subscribeoncreate, subscribeonreply and, schedule

/>

Description:
Creates an HTML input element for the desired user value. **Scripting Variables:**
none

Body Tags:
none

---

input.wizardfield.record

top

<IM:input.wizardfield.record
css=*"string"*
-
The unique id representing the WizardFieldObject

></IM:input.wizardfield.record >

Description:
Tag that provides the input html representing the current WizardFieldObject.

---

is.admin

top

<IM:is.admin
negate=*"non-scriptable string"*
-
Reverses the conditional of this tag. If it evaluates to "true" it will show the contents of this tag if the user is not an admin user. The default behaviour of this attribute is to be false.

></IM:is.admin >

Description:
Checks if the logged in user(if any) is an admin user. If it is, it displays the contents inside this tag, and if the user is not an admin user, it doesn't display the contents. Example Usage:

```
<IM:is.admin>
        HTML for admin users. In example: a link for the admin application
        <a href="http://www.someadminurl.com?<IM:logincredentials/>">Go To Admin</a>
</IM:is.admin>
<IM:is.admin negate="true">
        HTML for non admin users
</IM:is.admin>
```

---

is.inquirasearch.enabled

top

<IM:is.inquirasearch.enabled
negate=*"non-scriptable string"*
-
Reverses the conditional of this tag. If it evaluates to "true" it will show the contents of this tag if the InQuira Search is enabled. The default behaviour of this attribute is to be false.

></IM:is.inquirasearch.enabled >

Description:
Checks if the InQuira Search is configured

---

is.loggedin

top

<IM:is.loggedin
negate=*"non-scriptable string"*
-
Reverses the conditional of this tag. If it evaluates to "true" it will show the contents of this tag if the user is not logged in. The default behaviour of this attribute is to be false.

></IM:is.loggedin >

Description:
Checks if there is a logged in user in the session. If there is one, it displays the contents inside this tag, and if there is no user logged in it doesn't display the contents. Example Usage:
<IM:is.loggedin>
    HTML for logged in users
</IM:is.loggedin>
<IM:is.loggedin negate="true">

---

is.shoppingcart.option.selected

top

<IM:is.shoppingcart.option.selected
key*=*"string"*
-
The name used for the option.

value*=*"string"*
-

The value used for the option.

negate=*"non-scriptable string"*
-
Reverses the conditional of this tag. If it evaluates to "true" it will show the contents of this tag if the value is not present. The default behaviour of this attribute is to be false.

></IM:is.shoppingcart.option.selected >

Description:
Checks if a shopping cart option is selected by the user for a particular key and value. This tag must be embedded inside a `get.shoppingcart.record` tag

---

is.subscribed

top

<IM:is.subscribed
id=*"non-scriptable string"*
-
A desired prefix to use for all scripting variables created by this tag. There is no default id. Make sure all ids are unique

negate=*"non-scriptable string"*
-
Reverses the conditional of this tag. If it evaluates to "true" it will show the contents of this tag if the user is subscribed to the item. The default behaviour of this attribute is to be false.

userid=*"string"*
-
The userid for the user who's subscription we are checking.

type*=*"string"*
-
The type of subscription this is, Valid types are CHANNEL,CONTENT,FORUM,TOPIC.

recordid*=*"string"*
-
The recordid of the item to check for subscription.

categories=*"string"*
-
List of categories separated by plus(+)s signs

viewlocale=*"string"*
-
List of categories separated by plus(+)s signs

></IM:is.subscribed >

Description:
Checks if the currently logged in user is subscribed to a given item. If he is, we display the contents inside this tag

---

iterate.channel.child.records

top

<IM:iterate.channel.child.records
url=*"string"*
-
The url of the page we are currently viewing. DO NOT use, for internal application use only

maxpageitems=*"integer"*
-
Used by the paging mechanism, this defines the maximum items to display (iterate over) per page.

maxindexpages=*"integer"*
-
Used by the paging mechanism, this defines the maximum number of pages to display.

isoffset=*"true" or "false"*
-
The offset of the page we are currently viewing. DO NOT use, for internal application use only.

></IM:iterate.channel.child.records >

Description:
Iterates over content records associated with the current records.

Example Usage:

```
<!-- Get passed in record id form URL-->
<% String rec = request.getParameter("rec");%>

<!--Get handle to current record-->
<IM:get.channel.record recordid="<%=rec%>">
   <!-- Display current record title and body attribute for current record -->
   <IM.get.channel.attribute attribute="NEWS\TITLE"><br>
   <IM.get.channel.attribute attribute="NEWS\BODY"><br>
   Related Content:<br>
   <!-- Iterate channel child record -->
   <IM:iterate.channel.child.record id="relatedrecords">
      <!--Get handle to current record-->
      <IM:get.channel.record id="relatedrecord">
         <!-- Display title attribute for current record -->
         <IM.get.channel.attribute attribute="NEWS\TITLE">
         <!-- Provide hyperlink to same detail page for the related record -->
         <a href="index?page=detail&rec=<%=relatedrecord.recordid%>>Read more</a><br>
      </IM:get.channel.record>
   </IM:iterate.channel.child.record>
</IM:get.channel.record>
```

Scripting Variables:

{id}.faqsmaxitems - used internally for page batching
{id}.faqspagenumber - used internally for page batching
{id}.faqsoffset - used internally for page batching

Body Tags:

get.channel.record
get.channel.attribute

---

iterate.channel.parent.records

top

<IM:iterate.channel.parent.records
url=*"string"*
->

The url of the page we are currently viewing. DO NOT use, for internal use only

maxpageitems=*"integer"*

-

Used by the paging mechanism, this defines the maximum items to display (iterate over) per page.

maxindexpages=*"integer"*

-

Used by the paging mechanism, this defines the maximum number of pages to display.

isoffset=*"true" or "false"*

-

The offset of the page we are currently viewing. DO NOT use, for internal use only.

></IM:iterate.channel.parent.records >

Description:

Iterates over content records that associated the current record.

Example Usage:

```
<!-- Get passed in record id form URL-->
<% String rec = request.getParameter("rec");%>

<!--Get handle to current record-->
<IM:get.channel.record recordid="<%=rec%>">
   <!-- Display current record title and body attribute for current record -->
   <IM.get.channel.attribute attribute="NEWS\TITLE"><br>
   <IM.get.channel.attribute attribute="NEWS\BODY"><br>
   Parent Content Records:<br>
   <!-- Iterate channel child record -->
   <IM:iterate.channel.parent.record id="parentrecords">
      <!--Get handle to current record-->
      <IM:get.channel.record id="parentrecord">
         <!-- Display title attribute for current record -->
         <IM.get.channel.attribute attribute="NEWS\TITLE">
         <!-- Provide hyperlink to same detail page for the related record -->
         <a href="index?page=detail&rec=<%=parentrecord.recordid%>>Read more</a><br>
      </IM:get.channel.record>
   </IM:iterate.channel.parent.record>
</IM:get.channel.record>
```

Scripting Variables:

{id}.faqsmaxitems - used internally for page batching
{id}.faqspagenumber - used internally for page batching
{id}.faqsoffset - used internally for page batching

Body Tags:

get.channel.record

get.channel.attribute

iterate.channel.records

top

<IM:iterate.channel.records
dataset=*"string"*
-
Name of the dataset to iterate over.

id=*"non-scriptable string"*
-
Desired prefix to use for all scripting variables created by this tag. The default value is 'iterator'. If nesting channel-iterator tags, make sure all ids are unique

node=*"string"*
-
Path to desired node(s).

url=*"string"*
-
The url of the page we are currently viewing. DO NOT use, for internal application use only

maxpageitems=*"integer"*
-
Used by the paging mechanism, this defines the maximum items to display (iterate over) per page.

maxindexpages=*"integer"*
-
Used by the paging mechanism, this defines the maximum number of pages to display.

isoffset=*"true" or "false"*
-
The offset of the page we are currently viewing. DO NOT use, for internal application use only.

parentinlaw=

-
TODO

></IM:iterate.channel.records >

Description:
Iterate over all content records in a dataset.

Example Usage:

```
<!--Get channel data records using query-->
<IM:get.channel.data query="topnews" dataset="mynews">
<!-- Iterate channel record -->
<IM:iterate.channel.record dataset="mynews">
   <!--Get handle to current record-->
   <IM:get.channel.record id="news">
      <!-- Display title attribute for current record -->
      <IM.get.channel.attribute attribute="NEWS\TITLE">
      <!-- Provide hyperlink to detail page for current record -->
      <a href="index?page=detail&rec=<%=news.recordid%>>Read more</a><br>
    </IM:get.channel.record>
</IM:iterate.channel.record>
```

Scripting Variables:
{id}.faqsmaxitems - used internally for page batching
{id}.faqspagenumber - used internally for page batching
{id}.faqsoffset - used internally for page batching

Body Tags:
get.faq.question
get.faq.answer

---

iterate.dataform.answer

top

<IM:iterate.dataform.answer

></IM:iterate.dataform.answer >

Description:
Iterate over the answer records for a question in a dataform.

Example Usage:
Displaying the Form

```
<IM:form.dataform dataform="INQUIRIES" success="requestthank" error="requestthank">
   <IM:iterate.dataform.question>
      <IM:get.dataform.question.record>
         <b><IM:get.dataform.question/></b><br>
         <IM:iterate.dataform.answer>
            <IM:get.dataform.answer.record>
               <IM:input.dataform.answer/><br>
            </IM:get.dataform.answer.record>
         </IM:iterate.dataform.answer>
      </IM:get.dataform.question.record>
   </IM:iterate.dataform.question> <br>
   <input type="submit" value="Submit">
</IM:form.dataform>
```

Displaying Results

```
<IM:get.dataform.results name="nameofform" individual="false" aggregate="false">
  <IM:iterate.dataform.question>
      <IM:get.dataform.question.record>
         <b><IM:get.dataform.question/></b><br>
         <IM:iterate.dataform.answer>
            <IM:get.dataform.answer.record>
               <IM:get.dataform.answer/><br>
            </IM:get.dataform.answer.record>
         </IM:iterate.dataform.answer>
      </IM:get.dataform.question.record>
   </IM:iterate.dataform.question>
</IM:get.dataform.results>
```

Scripting Variables:
{id}.faqsmaxitems - used internally for page batching
{id}.faqspagenumber - used internally for page batching
{id}.faqsoffset - used internally for page batching

Body Tags:
get.dataform.answer.record
get.dataform.answer
input.dataform.answer

---

iterate.dataform.question

top

<IM:iterate.dataform.question

></IM:iterate.dataform.question >

Description:
Iterate over the question records in a data form.

Example Usage:
Displaying the Form

```
<IM:form.dataform dataform="INQUIRIES" success="requestthank" error="requestthank">
   <IM:iterate.dataform.question>
      <IM:get.dataform.question.record>
         <b><IM:get.dataform.question/></b><br>
         <IM:iterate.dataform.answer>
            <IM:get.dataform.answer.record>
               <IM:input.dataform.answer/><br>
            </IM:get.dataform.answer.record>
         </IM:iterate.dataform.answer>
      </IM:get.dataform.question.record>
   </IM:iterate.dataform.question> <br>
   <input type="submit" value="Submit">
</IM:form.dataform>
```

Displaying Results

```
<IM:get.dataform.results name="nameofform" individual="false" aggregate="false">
  <IM:iterate.dataform.question>
      <IM:get.dataform.question.record>
         <b><IM:get.dataform.question/></b><br>
         <IM:iterate.dataform.answer>
            <IM:get.dataform.answer.record>
               <IM:get.dataform.answer/><br>
            </IM:get.dataform.answer.record>
         </IM:iterate.dataform.answer>
      </IM:get.dataform.question.record>
   </IM:iterate.dataform.question>
</IM:get.dataform.results>
```

Scripting Variables:
{id}.faqsmaxitems - used internally for page batching
{id}.faqspagenumber - used internally for page batching
{id}.faqsoffset - used internally for page batching

Body Tags:
get.dataform.question.record
get.dataform.question
iterate.dataform.answer
get.dataform.answer.record

get.dataform.answer
input.dataform.answer

---

iterate.dataset

top

<IM:iterate.dataset
dataset*=*"string"*
-
Dataset variable for the data

id=*"non-scriptable string"*
-
A desired prefix to use for all scripting variables created by this tag. Ids must be unique

batchsize=*"integer"*
-
Desired Batch size for dataset

></IM:iterate.dataset >

Description:
Generic Tag that iterates over a list of records provided by the dataset.
Scripting Variables: - index = returns the current index of the iteration
- count = total number of objects provided by the dataset

---

iterate.emailsubscription.records

top

<IM:iterate.emailsubscription.records
userid=*"string"*
-
The userid for which to get the available subscriptions.

type=*"string"*
-
The type of subscriptions to fetch, Valid types are CHANNEL, CONTENT ,FORUM and, TOPIC. If this attribute is null all subscription types will be fetched

></IM:iterate.emailsubscription.records >

Description:
Tag that is used for iterating over the list of available subscriptions.

---

iterate.index

top

<IM:iterate.index

></IM:iterate.index >

Description:

---

iterate.message.records

top

<IM:iterate.message.records
url=*"string"*
-
The url of the page we are currently viewing. DO NOT use, for internal application use only

maxpageitems=*"integer"*
-
Used by the paging mechanism, this defines the maximum items to display (iterate over) per page.

maxindexpages=*"integer"*
-
Used by the paging mechanism, this defines the maximum number of pages to display.

isOffset=*"true" or "false"*

-

The offset of the page we are currently viewing. DO NOT use, for internal application use only.

talkbackdepth=*"integer"*

-

the depth of talkback to retrieve

contentid=*"string"*

-

guid of talkback record to iterate over

></IM:iterate.message.records >

Description:
Iterate over message records passed in with dataset.

Example Usage:
Displaying All Messages**:**

```
 <!-- get message data  -->
<IM:get.message.data dataset="data" topic="hammers"/>
<!-- iterate through messages in discussion records  -->
<IM:iterate.message.records dataset="data">
   <!-- get handle to individual record -->
   <IM:get.message.record>
      <!-- space message to show hierarchy and display message title and text-->
      <img src="images\spacer.gif" width="<% =message.depth %>" aligh="left"/><IM:get.message.title/> <br>
      <img src="images\spacer.gif" width="<% =message.depth %>" aligh="left"/><IM:get.message.text/> <br>
      <!-- display a reply link that passes the message record id to the reply page -->
      <a href="index?page=messagereply&rec=<%=message.recordid%>">Reply</a> <br>
   </IM:get.message.record>
</IM:iterate.message.records>
```

Replying to a Message:

```
<!-- declare page variable and load recordid parameter off url from previous page -->
<% String rec = request.getParameter("rec");%>
<!-- create HTML form for input fields -->
<IM:form.message parentid="<%=rec%>" success="successpage" error="errorpage">
   <!-- display input fields for title and text -->
   Title<br>
   <input.message.title><br>
   Text<br>
   <input.message.text><br>
   <!-- display submit button -->
```

<input type="submit" value="Submit">
</IM:form.message>

Scripting Variables:

Body Tags:
get.message.record
get.message.title
get.message.text
form.message
input.message.title
input.message.text

---

iterate.metadata.records

top

<IM:iterate.metadata.records
dataset*=*"string"*
-
A unique identifier for the dataset, this identifier is used by the iterate.channel.records and get.channel.template tags to access records that are retrieved.

></IM:iterate.metadata.records >

Description:
Tag for iterating over a dataset of metadata records.

---

iterate.node

top

<IM:iterate.node
node*=*"string"*
-
Xpath of node to be iterated

id=*"non-scriptable string"*

-
A desired prefix to use for all scripting variables created by this tag. There is no default id.

></IM:iterate.node >

Description:
Tag that iterates over a list of nodes defined in the xpath of a user.

---

iterate.search.portlets

top

<IM:iterate.search.portlets
order=*"string"*
-
The unique id representing PreviousResponses up to the wizard step id passed in

dataset*=*"string"*
-
A unique identifier for the dataset, this identifier is used by the iterate.dataset to step through all of the returned portlets. To retrieve the value passed in.

></IM:iterate.search.portlets >

Description:
Iterate over the list of the portlets

---

iterate.shoppingcart.record.options

top

<IM:iterate.shoppingcart.record.options
key=*"string"*
-
If you specify a key, this tag will only iterate over options with that key

></IM:iterate.shoppingcart.record.options >

Description:
Iterates over the selected options for a particular shopping cart record

---

iterate.shoppingcart.records

top

<IM:iterate.shoppingcart.records
dataset*=*"string"*
-
Dataset variable for the shopping cart

></IM:iterate.shoppingcart.records >

Description:
Iterates over the shopping cart data

---

iterate.subscription.records

top

<IM:iterate.subscription.records
view=*"string"*
-
The reference key of the desired view.

dataform=*"string"*
-
The name of the data form from which get the available subscriptions.

></IM:iterate.subscription.records >

Description:
Tag that is used for iterating over the list of available News Letter.

---

iterate.wizard.previous.responses

top

<IM:iterate.wizard.previous.responses
stepid*=*"string"*
-
The unique id representing PreviousResponses up to the wizard step id passed in

></IM:iterate.wizard.previous.responses >

Description:
Iterate over the list of the Process Wizard's previous responses

---

iterate.wizardform.fields

top

<IM:iterate.wizardform.fields

></IM:iterate.wizardform.fields >

Description:
Iterate over the ProcessWizard form's WizardFieldObjects

---

logout

top

<IM:logout

></IM:logout >

Description:
This tag invalidates the current session. To use just place this tag in the logout page

manage.content

top

<IM:manage.content
contentid=*"string"*
-
If a contentid is provided with the login credentials, the user is directed directly to the associated content record for editing.

category=*"string"*
-
If a category reference key is passed in, it will be preselected in the content edit screen and it will hide the category section.

success=*"string"*
-
fully qualified URL to return the user to if the update was successful

error=*"string"*
-
fully qualified URL to return the user to if an error occurs

channel=*"string"*
-
Reference Key for ContentChannel. This parameter is required if adding content

view=*"string"*
-
This is a view reference key. If this property is set and the action is add, then the content will be assigned to this view if the logged in user has enough permissions

action*=*"string"*
-
ADD - EDIT - DELETE. Desired action to be performed. This is a required parameter that defines what action should be performed.

localecode=*"string"*
-
used for showing translated version of doc

/>

Description:
Returns A link to the admin tool used to manage content . It should specify if the action is ADD, DELETE or MODIFY. The result is a link with encrypted user credentials used for login in the admin application. If the contentid and success and error attributes are passed, the user will be brought directly to that content record for editing
Usage Example:
<a href="http://www.applicationurl.com?<IM:user.admin.link/>">Go To Admin

---

node.count

top

<IM:node.count
attribute*=*"string"*
-
Xpath to the nodes

id=*"non-scriptable string"*
-
id of scripting variable

/>

Description:
Counts the number of nodes returned by an xpath

---

pdf

top

<IM:pdf
filename=*"string"*
-
Name for the generated PDF file

fontsize=*"string"*

-

Base font size for the generated PDF Document

font=*"string"*

-

Sets the font of the document. Allowed values: Arial, Courier, Helvetica, Monospace, Sans-Serif, Serif, Symbol, Times

></IM:pdf >

Description:
This tag converts everything that is enclosed to a pdf document.

---

save.content.attribute

top

<IM:save.content.attribute
attribute*=*"string"*

-

Attribute to be modified. To edit display start and end dates, or event start and end dates, you must specify one of the following keywords and a date mask DISPLAYSTARTDATE, DISPLAYENDDATE, EVENTSTARTDATE, EVENTENDDATE

value*=*"string"*

-

Value to be saved

mask=*"string"*

-

Date mask for Content Dates.

/>

Description:
Tag that specifies the attribute and value to be modified on a content record using the save.content.data tag. This tag should always be embeded inside a `save.content.data/code> tag`

---

save.content.data

top

<IM:save.content.data
contentid*=*"string"*
-
Content id of the content to be modified.

publish=*"true" or "false"*
-
Flag that determines if the modified content record should be published or not

></IM:save.content.data >

Description:
Tag that automatically saves content data into a content record. This tag is used in conjunction with the `save.content.attribute` tag

---

save.user.attribute

top

<IM:save.user.attribute
attribute*=*"string"*
-
The path to save the value

value*=*"string"*
-
The value to be saved.

></IM:save.user.attribute >

Description:
This tag will save a value to the passed in attribute path. If this tag is embedded within a get.user.record tag, it will save the property to the user specified in that tag, otherwise it will save the property to the logged in user

---

set.content.casenumber

top

<IM:set.content.casenumber
documentid="string"
-
Set the case number for the supplied documentid

recordid="string"
-
Set the case number for the supplied recordid

casenumber*="string"
-
The value to set the case number for the content record

casedescription*="string"
-
The value to set the case description for the content record

caseincident*="string"
-
The value to set the case incident for the content record

></IM:set.content.casenumber >

Description:
Tag that sets a case number for a content record

---

set.search.term

top

<IM:set.search.term
attribute="string"
-
If performing a attribute level search, set the desired attribute here.

value=*"string"*

-

The actual text to search.

/>

Description:
Tag that is used in conjunction with the get.search.data tag. This tag sets the text to be searched.

---

set.shoppingcart.status

top

<IM:set.shoppingcart.status
cart*=*"string"*

-

Refrence key for shopping cart.

workflow=*"string"*

-

Reference key of the workflow step you wish to move the transaction to.

status*=*"string"*

-

This will be the new status for the transaction. Allowed values are APPROVED, REJETED, and OPEN. If you are using an authorization gateway, you may want to set the status to APPROVED after a transaction has been completed and approved, set to REJETED to represent that the authorization system did not approved the transaction. If you are not using am authorization system, you may just set it to OPEN, representing that the transaction has been completed and is in the workflow system ready for further actions.

code=*"string"*

-

String representing a code for this transaction. This could be used to store confirmation, rejected codes, or any custom code.

amountbilled=*"string"*

-

String that represents the total amount billed for a transaction. This is an optional attribute for transactions that actually involve monetary transactions.

terminate=*"true" or "false"*

-
If evaluates to true, the shopping cart transaction will be removed from the session or the cookies on the user's computer. You may want to set this to true after a transaction has been approved so that the user's cart gets emptied.

transactionid=*"string"*
-
Optional field for setting the transactionid if known. This attribute will get the transaction directly from the database instead of the user's session, or cookie.

></IM:set.shoppingcart.status >

Description:
Tag that automatically changes the status of a shopping cart transaction and places the transaction at the passed in workflow step defined for the shopping cart. The allowed values for statuses are APPROVED, REJECTED, and OPEN

---

set.user.keyvalue

top

<IM:set.user.keyvalue
id*=*"string"*
-
key to request the value for

value*=*"string"*
-
value to set

recordid=*"string"*
-
The userid for the user we are setting the key value pair for

login=*"string"*
-
The login id for the user we are getting the key value pair for

></IM:set.user.keyvalue >

Description:
Tag that sets or updates a key/value pair

---

sitemap

top

<IM:sitemap
pagename*=*"non-scriptable string"*
-
The name of the page that will be used in url's by the Index servlet. i.e. http://www.company.com/index?page=pagename

requireslogon=*"true" or "false"*
-
Determines if the page requires the user to be logged in prior to viewing the page

ssl=*"non-scriptable true" or "false"*
-
Determines if the page should be displayed using SSL (https)

logonpage=*"non-scriptable string"*
-
Name of the page that the user will be redirected to if they have not been logged in yet, this name should be the name of another Sitemap page entry

default=*"non-scriptable true" or "false"*
-
Determines if the current page is the default page that will show if there is an error finding the specified page.

namespace=*"string"*
-
Allows designer to specify the namespace that this page will respond to. If set, only the specific namespace (IMe sensitive) will contain the page.

/>

Description:
Creates a Sitemap entry used to specify security and login requirements for a page. This tag is only needed on JSP pages that are not components of other pages. For example, Home.jsp may include several other files, only Home.jsp will need this tag, the included files do not need a sitemap tag. The Sitemap serves a couple of

purposes: first it provides a mechanism to protect the underlying structure of your site, and secondly it gives a shorter more user friendly URL for bookmarking.
To display the sitemap append &sitemap to your URL. i.e. http://www.company.com/index?page=home&sitemap

---

template.definition

top

<IM:template.definition
template*=*"string"*
-
Sitemap name for the template to use

></IM:template.definition >

Description:
Creates a template definition

---

template.get

top

<IM:template.get
name*=*"string"*
-
Name of template entry

></IM:template.get >

Description:
Gets a template definition entry for content

---

template.put

top

<IM:template.put
name*="string"
-
Sitemap name for the template to use

direct="string"
-
Whether to include the code as JSP or as part of the template

content*="string"
-
JSP file include or HTML to use for the template

></IM:template.put >

Description:
Adds a template definition entry

---

timer

top

<IM:timer
id="string"
-
unique id for the code block

></IM:timer >

Description:
This tag is used to benchmark the performance of a code block

---

update.content.metric

top

<IM:update.content.metric
recordid*=*"string"*
-
Record id of the content to be updated.

metric=*"string"*
-
Reference key of the custom metric to be used.

></IM:update.content.metric >

Description:
Tag that automatically updates the view content for a particular content record. It increments the view count value by one, and if it is the first time viewed it updates it to one. If no metric reference key is passed in, it assumes you want to update the DEFAULT metric.

---

user.admin.link

top

<IM:user.admin.link
contentid=*"string"*
-
If a contentid is provided with the login credentials, the user is directed directly to the associated content record for editing.

view=*"string"*
-
This is a view reference key. If this property is set and the action is add, then the content will be assigned to this view if the logged in user has enough permissions

success=*"string"*
-
fully qualified URL to return the user to if the update was successful

error=*"string"*
-
fully qualified URL to return the user to if an error occurs

/>

Description:
Returns the credentials for the logged in user encrypted. These credentials are used for login in the admin application. If the contentid and success and error attributes are passed, the user will be brought directly to that content record for editing
Usage Example:
<a href="http://www.applicationurl.com?<IM:user.admin.link/>">Go To Admin

---

user.input.remove.file

top

<IM:user.input.remove.file
attribute*="string"
-
Path to custom user attribute.

value="string"
-
Initial value of the input field. Note : if nothing is entered for the value, the value will get default values based upon the information in the database.

css="string"
-
CSS Class to be used for the input field

/>

Description:
Creates an HTML input element for removing a file from the server for the specified name stored in the attribute.

---

get.contenthistory.data

top

<IM:get.contenthistory.data
dataset*="string"
-

A unique identifier for the dataset. This identifier is used by the iterate.records tag to access records that are retrieved.

documentid=*"string"*
-
A content documentID. Prepend with 'S:' for unpublished content. A documentID OR recordID is required.

recordid=*"string"*
-
A content recordID. Prepend with 'S:' for unpublished content. A recordID OR documentID is required.

localecode=*"string"*
-
Returns records relating to content given localecode.

login=*"string"*
-
Returns records relating to a given user login.

version=*"string"*
-
Version to return history for.

maxversions=*"string"*
-
Number of versions before version to retrieve history for.

/>

Description:
Retrieves a list of type ContentHistoryObject

---

get.contenthistory.record

top

<IM:get.contenthistory.record
id=*"non-scriptable string"*
-
A desired prefix to use for all scripting variables created by this tag.

></IM:get.contenthistory.record >

Description:
Tag that gets a content history record from an iteration