Consent Management

Overview

Compliance regulations are changing fast, and growing in complexity and scope. In addition, users have come to expect from brands a high level of transparency and control over their user data. Companies that fail to address this, and to supply their users with a clear flow for agreeing to the way their data is handled, risk customer abandonment and heavy fines. Gigya helps you comply with the increasingly complex legal requirements around user data privacy. The Consent management dashboard allows you to create and manage versions of different types of user consent regarding user data and privacy.

> While Gigya offers a full suite of solutions designed to help clients comply with applicable data privacy laws, it is the clients' responsibility to comply with its obligations under such data privacy laws. Please consult with your legal team regarding such data privacy laws prior to implementation of the Gigya suite of solutions.

Watch an Instructional Video

If you have an SAP logon, you can watch the following instructional videos:

- Implementing SAP Customer Consent
- Implementing SAP Customer Consent without SAP Customer Identity

About Consent Statements

Gigya offers configuring three types of consent statements:

- Terms of service: it is mandatory for users agree to these terms in order to enjoy your site or app's services,
- Privacy policy: it is mandatory for users to agree to these terms in order to enjoy your site or app's services.
- Other consent statements: these can be configured to mandatory or not.

Present consent statements on lite and full registration screens, as a prerequisite for receiving site services. The consent statement can be displayed in a local language, and may include the reason why personal data is being collected (**Purpose**), and a link to the document to which they are agreeing. User consent is captured and saved in the consent vault, including consent for mandatory terms, non-mandatory terms, and communication preferences.

Once a consent statement is activated, it is a required field in your site schema as a condition for registration, meaning a new user cannot register to your site without at least viewing this policy. In other words, the registration flow has to pass to Gigya either "true" or "false" for the **isConsentGranted** parameter, and cannot pass "null". In addition, two types of statements, **terms of service** and **privacy policy**, are mandatory for users to complete their registration. This means that users attempting to register to your site, have to willingly agree to any active terms and policies that apply to that site (**isConsentGranted** has to be true for the user to complete registration). In addition, users cannot withdraw this consent and remain registered to your site. If a user wishes to withdraw their consent, they should be deleted from your site. If that user tries to register again to your site, they will go through the registration process like a new user.

In addition, privacy regulations require that you enable users to access and review the agreements that they have signed when they registered to your site, and withdraw their consent from agreements that are not a prerequisite for using your site services.

Therefore, assuming that you are using Gigya's RaaS screens, it is highly important that you add to all the following screens a separate checkbox for each of the relevant statements:

- Registration
- Registration Completion
- Privacy

Validating the User's Consent Status

Consent management includes a built-in mechanism for ensuring that users who are logged in to your site, have a valid consent in place to the relevant mandatory statements. For example:

- A new user registers to your site but does not agree to the terms of service. The user will not be able to submit the registration screen.
- A user registered to your site before the consent module was implemented and has no valid consent statements associated with their account. The next time they log in, the "Registration Completion" screen will be displayed, requesting the relevant agreements.
- A user has agreed to your site consent statements, and is still logged in, but the site admins have changed the version of one of the consent statements. During their logged-in session, the "Registration Completion" screen will be displayed, requesting their re-consent.

- When the Registration Completion screen opens, if the user had an active session, that session is terminated.
- If a user closes ("X") the Registration Completion screen without agreeing to the new terms, an accounts.logout and socialize.logout event is fired, and the user is logged out of the site.

To ensure these scenarios take place, you must make sure to include the relevant consent statements on all registration and registration completion screens, and to set a value to the **verifyLoginInterval** parameter in the Global Configuration.

See below to learn about:

- Customizing the Registration Completion screen to display information conditionally: for example, to display different text based on whether the user has never given consent to these terms, or the version has changed and they now need to re-consent.
- Version control
- Syncing information to downstream applications using IdentitySync (Gigya's data transport solution), based on users' consent status.

Customizing Consent-Related Data

The consent interaction between the user and your site, is captured and recorded to Gigya’s database (in a JSON object known as the Preferences Object). This interaction includes several fixed components, such as the type of consent statement, the version number, whether the statement is mandatory or not, etc. Because of the importance placed on the validity and quality of the captured data, naturally these data points cannot be altered with ease.

In addition, Gigya provides you with several options for flexibly customizing the consent record to your needs. These customization options include:

- Custom data: Added to the schema of the consent object. Allows you to add any additional information to the object, such as the legal entity that maintains it, the admin who created it, or for adding a common value to consent records.
- Tags: Appended to a specific consent interaction. For example, if the same consent statement is added to two different screens (registration completion and profile update), the tags can indicate the screen on which the user granted their consent, i.e., to which interaction the consent corresponds.
- Entitlements: An addition or appendix to the consent interaction. An easy way to ask for additional input from a user (additional to consent they have already given), without having to trigger the re-consent process. Users can only grant entitlements if they have agreed to the original consent statement.

The following table compares between these features and their capabilities:

	Custom Data	**Tags**	**Entitlements**
Adds data to	Consent object schema	Consent interaction	Consent interaction
Useful for adding	Metadata	Context	Content
Depends on consent status?	No	No	Yes
Defined in	Consent dashboard, setSchema	UI Builder, profile update (setAccountInfo), registration process (accounts.register)	UI Builder , profile update (setAccountInfo), registration process (accounts.register)
Reflected in	Consent Vault, Identity Access, get Schema, getAccountInfo	Consent Vault, Identity Access, getAccountInfo	Consent Vault , Identity Access , getAccountInfo
Saved as	Key-value pairs	Array of strings. New values overwrite existing ones.	Array of strings. New values are appended to existing ones.* * See below for a more detailed explanation.
Can be edited by end user?	No	Usually not	Yes

Using This Guide

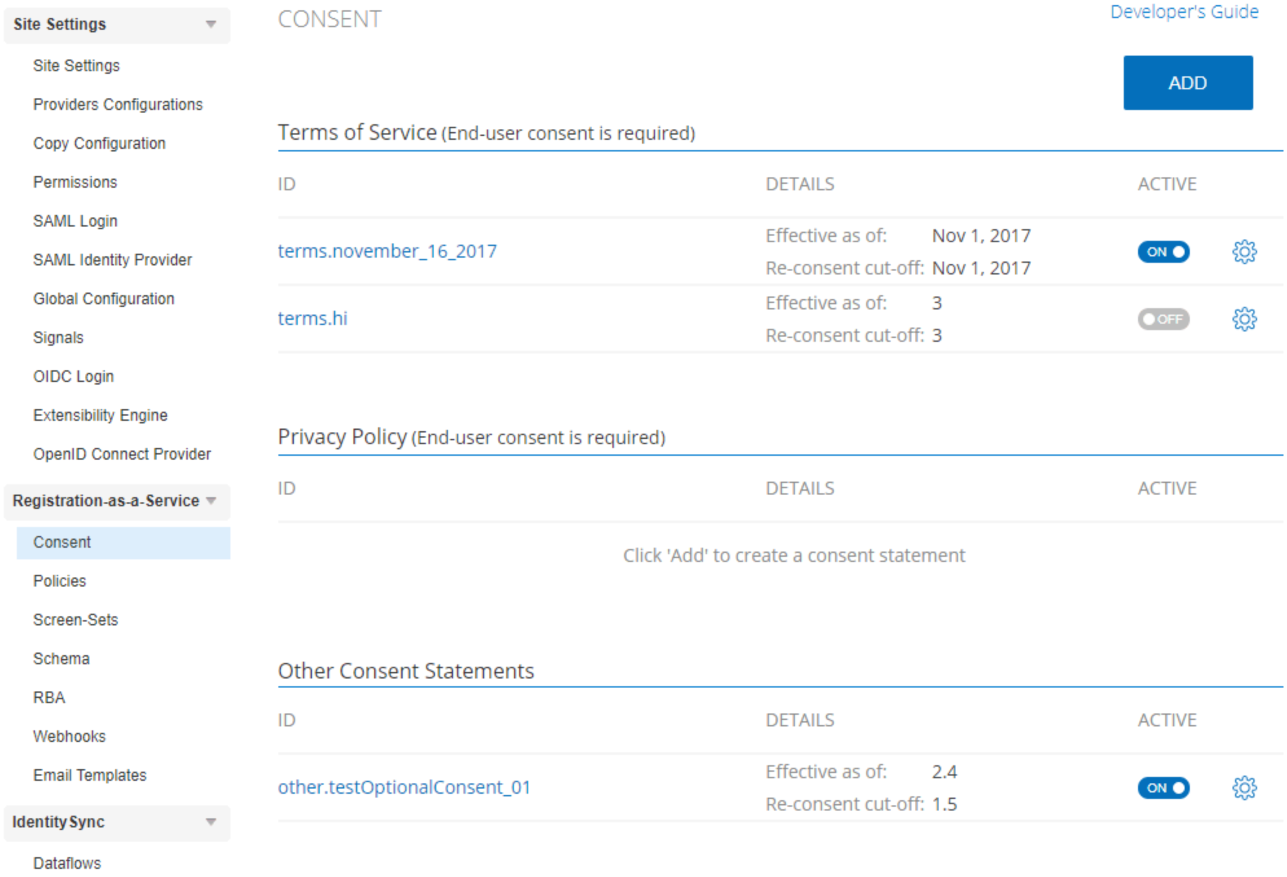
The following guide walks you through the steps of creating a new consent statement, adding it to a Gigya screen using the UI Builder, and managing consent version.

For more information on Gigya screens and screen-sets, see Screen-Sets.

The data structure of the consent statement is explained here.

Create a New Consent Statement

1. Go to the Consent dashboard in your Gigya Console.



Terms of Service (End-user consent is required)

ID	DETAILS	ACTIVE
terms.november_16_2017	Effective as of: Nov 1, 2017 Re-consent cut-off: Nov 1, 2017	ON
terms.hi	Effective as of: 3 Re-consent cut-off: 3	OFF

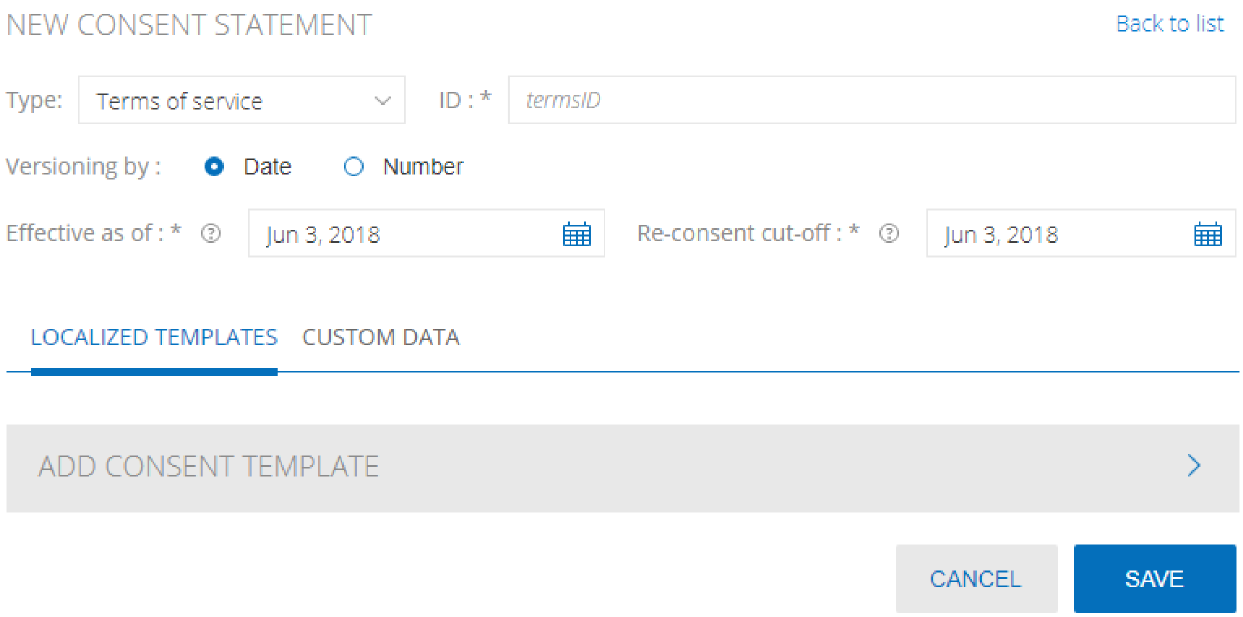
Privacy Policy (End-user consent is required)

ID	DETAILS	ACTIVE
Click 'Add' to create a consent statement		

Other Consent Statements

ID	DETAILS	ACTIVE
other.testOptionalConsent_01	Effective as of: 2.4 Re-consent cut-off: 1.5	ON

2. Click the **Add** button to create a new statement.



NEW CONSENT STATEMENT

Type: Terms of service

ID : * termsID

Versioning by : ● Date ○ Number

Effective as of : * Jun 3, 2018

Re-consent cut-off : * Jun 3, 2018

3. Select the **Type** and enter an ID for the new statement.

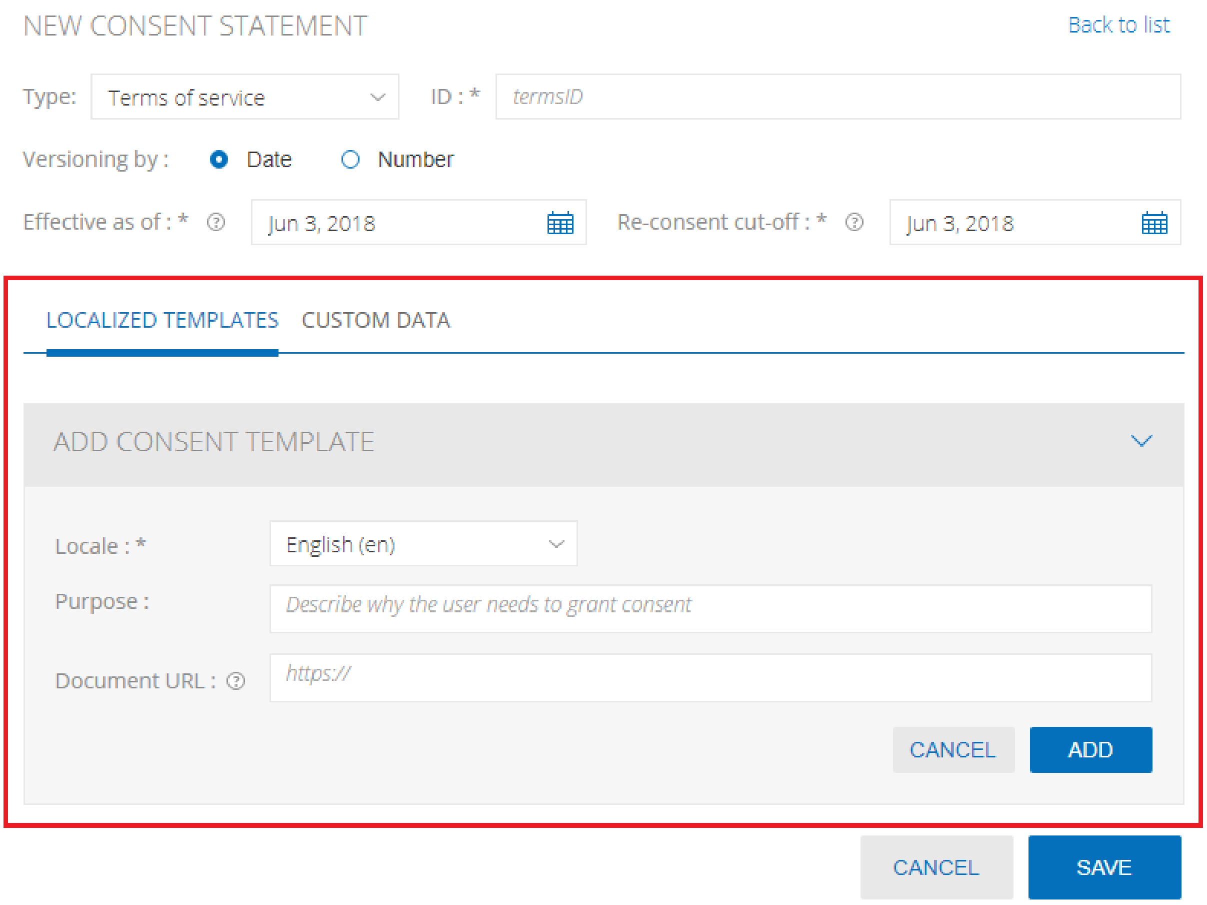
> The fixed prefix for terms of service is **terms**, and for privacy policies it's **privacy**.

4. Under **Versioning by**, select whether to use **Date** or **Number** and input the appropriate dates or version for this consent.

> Once you save the new statement, you cannot change this definition.

5. You can add a localized template, that includes a locale, the purpose of this statement, and a link to the document to which the user is agreeing. To do so, select **Add Consent Template** and enter the required details:
   a. Locale
   b. Purpose: The purpose of this consent, this will be visible to the end user.
   c. Document URL: The URL of the PDF version of this consent statement (must be HTTPS).

   > Any Document URLs provided must be persistent and the document must be available on the defined URI for as long as required by the country of residence of the end-user. It is the client's responsibility to maintain accurate records of any Consent Templates that were agreed to by the end-user.



NEW CONSENT STATEMENT

Type: Terms of service

ID : * termsID

Versioning by : ● Date ○ Number

Effective as of : * Jun 3, 2018

Re-consent cut-off : * Jun 3, 2018

ADD CONSENT TEMPLATE

Locale : * English (en)

Purpose : Describe why the user needs to grant consent

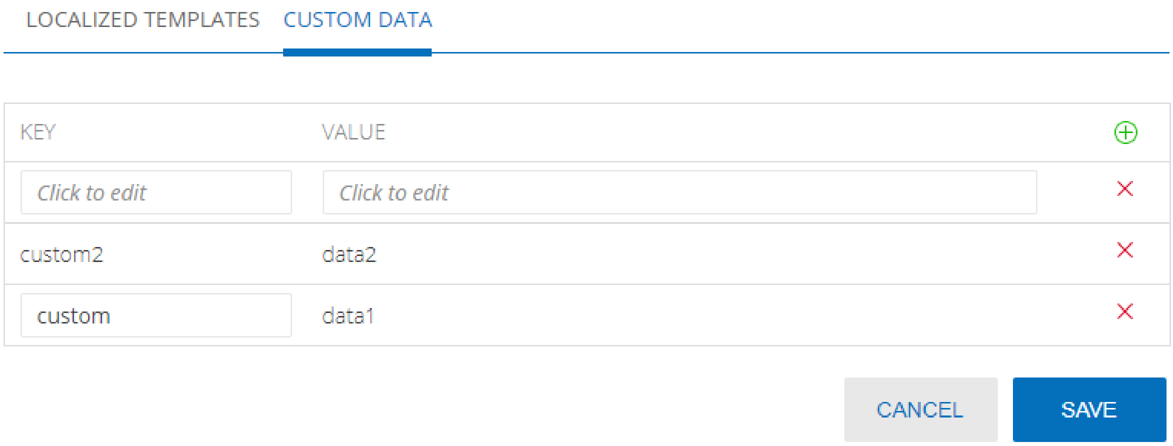
Document URL : https://

d. Click **Add**.
e. Repeat for any other supported locales that you offer.

> The legal statement and purpose will be displayed for the relevant end-user in the Consent History tab of Identity Access and in the Consent Vault, but not be visible on the user account via accounts.getAccountInfo.

6. You can add **Custom Data** (custom key-value pairs) to the consent statement. The custom data will be available on the account (when calling accounts.search or accounts.getAccountInfo), and will be audited in the consent vault. These pairs will be saved to the Preferences Schema Object.
   a. Click the **+** button
   b. Enter the key and the value. Both will be saved in string format.

> The maximum number of characters for the key is 20, and 256 for the value. The maximum number of custom key-value pairs per consent statement is 50.



KEY	VALUE
Click to edit	Click to edit
custom2	data2
custom	data1

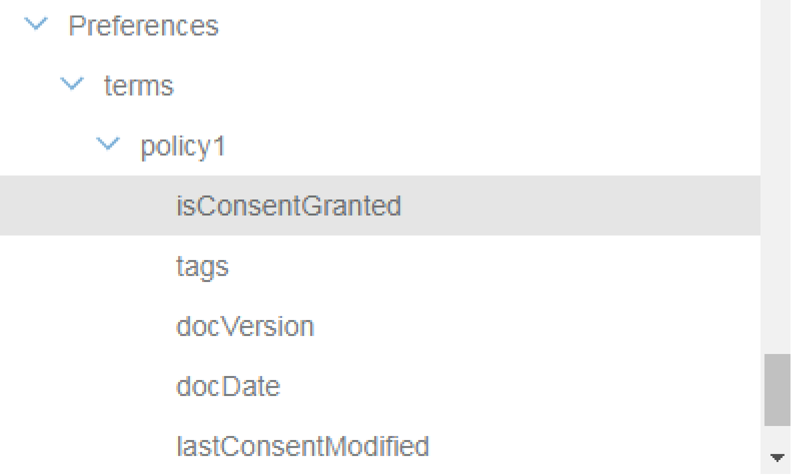
7. **Save** the new statement.

Add the Statement to a Screen

This section assumes you are using Gigya's screens, and are using the UI Builder to maintain them. Note that **any** consent statement that is set to "active" for your site, including a non-mandatory consent statement, has to be included in your registration and registration completion screens. Otherwise, users will not be able to complete their registration.

Registration Screens

1. Go to the Screen-Sets page in Gigya's Console.
2. Open the relevant screen-set in the UI Builder (e.g. the **Default-RegistrationLogin** screen-set, that contains both the Registration and the Complete Registration screens).
3. Select the screen from the list of screens on the left.
4. If the screen you are editing contains the default "terms of use" checkbox, delete or edit that checkbox so that it will not be mapped to **data.terms**. You can see the field to which a checkbox is mapped, under **Mapped Field** in the **Properties** pane on the right-hand side.
5. Under **Controls**, drag a checkbox into the canvas. Under **Properties**:
   a. Open the **Mapped Field** dropdown and scroll down to the **Preferences** namespace.
   b. Locate the relevant consent statement, and map the checkbox to the **isConsentGranted** field:



   c. Still under **Properties**, revise the label as necessary.

> If you are using Consent Templates, and you have configured at least one locale, you can use a placeholder to automatically display the **Purpose** in the user's locale, if a template exists. To use the placeholder, in the Label field of the checkbox's Properties enter the following, where the **terms.sampleConsentTerms** is the name of your consent

item.

```
{{schema.preferencesSchema.fields['terms.sampleConsentTerm
s'].legalStatements[screenset.lang].purpose}}
```

Or, if you want to place a link to the consent statement, you can use the following sample. If you are not using the UI Builder, add a "data-binding" attribute set to "true". This is added automatically when using the UI Builder:

```
<a target="_blank" data-binding="true"
href="{{schema.preferencesSchema.fields['terms.sampleConse
ntTerms'].legalStatements[screenset.lang].documentUrl
}}">click to view</a>
```

d. Before you save your changes, make sure the **Checked by Default** drop-down is set to **No**. Otherwise, you may risk a breach of regulatory compliance.
e. Save your changes.
f. Repeat the process for the **Registration Completion** screen and any other screens you require consent from the user.

Display Conditionally in Registration Completion Screens

As a best practice, in Registration Completion screens, you should display to users only the statements that they have not yet signed, so as to prevent confusion and misunderstandings. To do so:

1. Open the Registration Completion screen in the UI Builder.
2. Select the relevant checkbox, that is mapped to a consent statement.
3. In the Properties pane, under **Visible When**, enter a JavaScript expression that, when evaluated to 'true', will cause the field to be displayed. For example, if you have a consent statement called "myConsent", you can use the following:

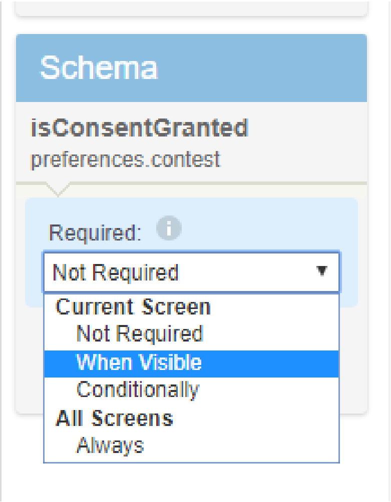
   ```
   !preferences || !preferences.myConsent ||
   !preferences.myConsent.isConsentGranted
   ```

   This will cause the checkbox element to display only if the user has never granted consent to these terms (isConsentGranted=false).
4. Select the **Keep visible** checkbox, to prevent the field from disappearing once the user accepts the terms.
5. Repeat the process for a different checkbox where the user *has* granted consent to these terms (isConsentGranted=true). This will display if the version of the terms has changed, and the user needs to re-consent.

Lite Registration Screens

When users perform "lite" registration to your site, many of the schema validations that have to do with registration are overriden. For this reason, to ensure that only users who have consented to the relevant terms can perform lite registration, the consent checkbox should be defined as mandatory when visible on the screen:

1. Open the Lite Registration screen in the UI Builder.
2. Drag a checkbox element to the screen.
3. Under **Properties**, map the checkbox to the "isConsentGranted" property of the relevant consent statement.
4. Provide the Label you wish to display to users.
5. Under the **Required** property (in the **Schema** section), select "When Visible". This means that the users cannot submit the screen without agreeing to these terms.



6. Repeat for any other consent statement you wish to enforce on this screen.
7. Save your changes.

Preference Management: Privacy Screen

The Privacy screen (that is in the ProfileUpdate screen-set) displays to registered users the details of the consent statements to which they agreed. To reflect to the user the statement to which they consented and the version and date that they did so:

1. Open the Privacy screen in the UI Builder.
2. Drag the **Consent** widget to the canvas from the left menu (under **Widgets**).
3. In the **Properties** pane, map the widget to the relevant consent object.
4. Reword the label as needed.

> The default Privacy screen already contains two consent widgets by default. If you are using these, you need only map them to the consent object and change the label.

5. If your site includes a consent statement that is not mandatory, add a checkbox to the Privacy screen that is mapped to the **isConsentGranted** field, so that they can grant or withdraw their consent.
6. Save changes.

Adding Tags

You can add a set of tags that will be passed together with the record of the consent interaction. These tags will be visible in the Consent Vault (and in Identity Access) for the captured consent.

To add tags:

1. Drag a Metadata control to the screen.
2. Map the control to the **tags** field of the relevant consent statement.
3. Under **Value format**, select **"**JavaScript expression".
4. In the **Value** field, add the tags as needed. For example, add an array of tags according to the following example:

```
["tag1","tag2","tag3"]
```

5. Save changes.

> **Notes**
>
> - In the UI Builder you may also save the tags as **Value format** *string* of the **Metadata** control, however, you must be sure to **not**

include any quotes around the individual tags, or your tags will be saved with the quotes as part of the tag name, i.e.,

```
Saving as a string: "hasCats", "hasDogs",  "hasFish"

results in:
"tags": [
      "\"hasCats\"",
      " \"hasDogs\"",
      "  \"hasFish\""
]
```

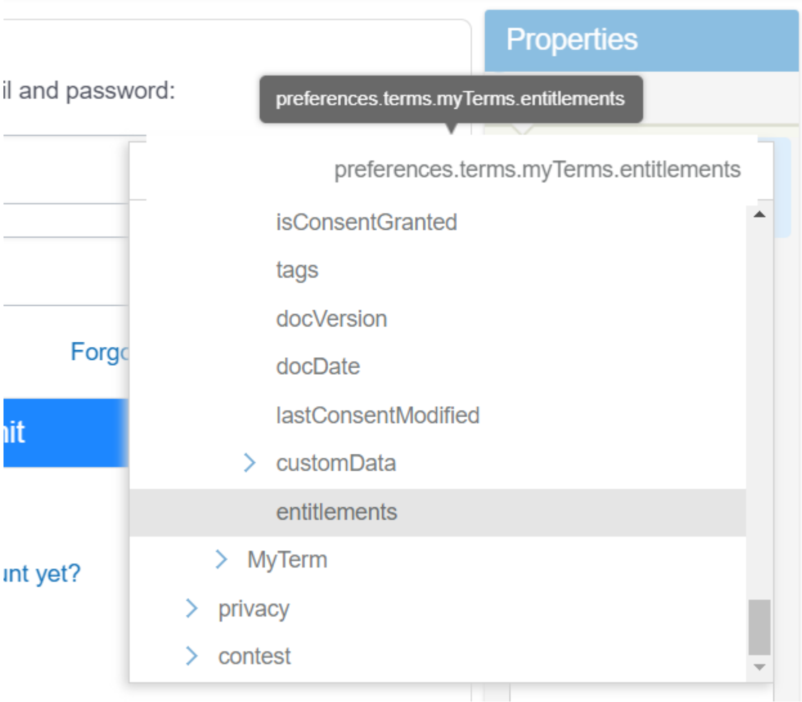
- Tags do not append. Whenever you attach tags to a consent interaction, the previous tags are replaced with the new ones.
- Tags may be attached to a consent interaction only if the consent status has changed (i.e., the value of **isConsentedGranted**).

Adding Entitlements

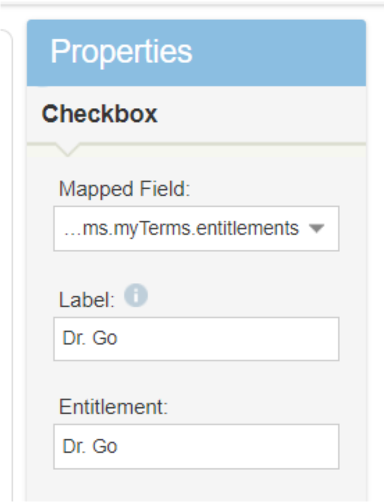
You can add an 'entitlements' property, in which to save a list of entitlements to which the user has granted consent. For example, if a user has granted consent to a medical clinic to view their personal data, they can then check the doctor or doctors who are entitled to view it. This list can later be exported to a third-party app that gives permissions to the medical staff to view the file.

To add an entitlements definition to a screen:

1. Open the relevant screen in the UI Builder.
2. From the list of controls on the left, drag a "Checkbox" element and drop it in the desired place in the screen.
3. In the **Properties** pane on the left, map the element to the "entitlement" property of the relevant consent statement:



4. Under **Entitlement**, type the entitlement.
5. Give the field a meaningful **Label**.



Properties

Checkbox

Mapped Field: …ms.myTerms.entitlements

Label: Dr. Go

Entitlement: Dr. Go

6. Save your changes.

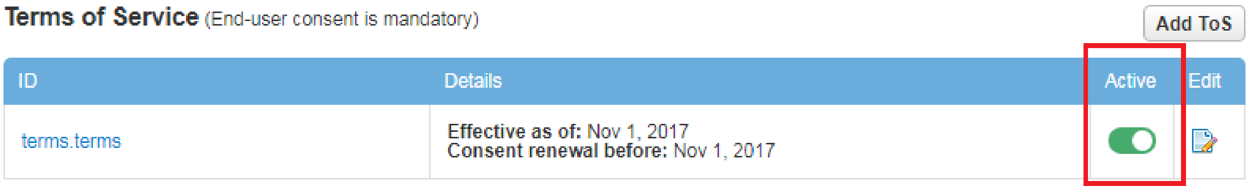
Notes

- The entitlement property alters the markup, so that when using Gigya's screens, the Web SDK automatically sends the entitlements as an array. API-based implementations that do not use the Web SDK, should set up some function that does this manually.
- The entitlement checkbox will appear disabled in the user-facing screen, unless users first agree to the relevant consent statement.
- If a user changes the entitlements to which they consent, this action will be recorded in the Consent Vault with the action noted as "Changed". However, it is not considered a re-consent.
- Entitlements sent from a screen are appended to existing entitlements saved to the user's account. However, note that if a checkbox displayed on the screen is mapped to an entitlement, whatever value is passed from the screen will override the existing value. For example, if a user previously agreed to entitlement "A", and in their profile update screen uncheck the "A" entitlement and save, the "A" entitlement will be deleted from their account.

Activating a Consent Statement

Consent statements can be activated in one of two ways:

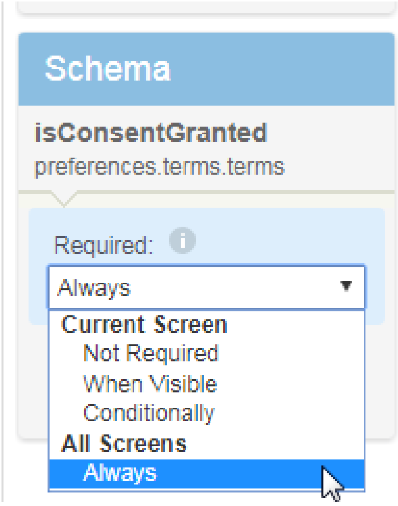
- In the **Consent** dashboard, using the **Active** toggle:



Terms of Service (End-user consent is mandatory)

ID	Details	Active	Edit
terms.terms	**Effective as of:** Nov 1, 2017 **Consent renewal before:** Nov 1, 2017	●	✎

- In the UI Builder **Properties**, by setting the field to required on all screens:



Whichever way you choose to activate the statement, you should ensure beforehand that the consent object is properly mapped to your screens, as explained above. Otherwise, users cannot complete their registration.

Refresh Interval

You can set up a global configuration for your site that requests users to re-consent to site terms after a specified period of time.

To do so:

1. Open the Global Configuration setting in Gigya's Console.
2. Add a line for the **refreshInterval** param, and specify (in numbers) the number of days since consent was previously given before requesting renewed consent.

> For more information, see Global Configuration

Shared Consent in Site Groups

Consent configuration differs in some respects when configuring site groups.

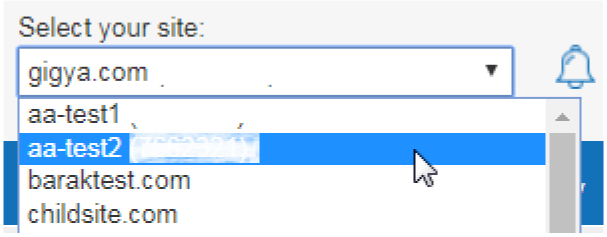
- Consent statements are created and configured only on the parent site (including version control), and activated or de-activated for individual child sites.
- The configuration of a consent statement (including the active version) is inherited by the child sites in the group, and may be overriden per individual site.

  > Once you change the configuration of a consent statement on a child site, inheritance from the parent is broken.

- Some sites may require that users agree to a separate set of terms and policies, or an additional set, on top of the shared consent statements. See below for configuration guidelines.
- When sites belong to the same SSO segment:
  - If the terms of service and privacy policies are identical for all sites in the segment, the terms should reflect to users that they are registering to the entire brand.
  - If sites in the same segment do not share or only partially share the consent statement, take care to configure a separate statement for each site that requires it (including presenting the option to accept the statement in the registration completion screen). For example, if sites A and B share the same terms of service, and site B requires additional terms, configure one statement that is active for site A, and another that is active for B only. When a user registers to site A, they will be asked to consent to the shared terms. When they navigate to site B, they will be prompted to agree to the individual "site B" terms. On the other hand, if a user registers to site B first, then navigates to site A, they will be logged in seamlessly.

To configure consent in site groups:

1. In the Consent page of Gigya's Console, make sure the parent site is selected in the site selector.



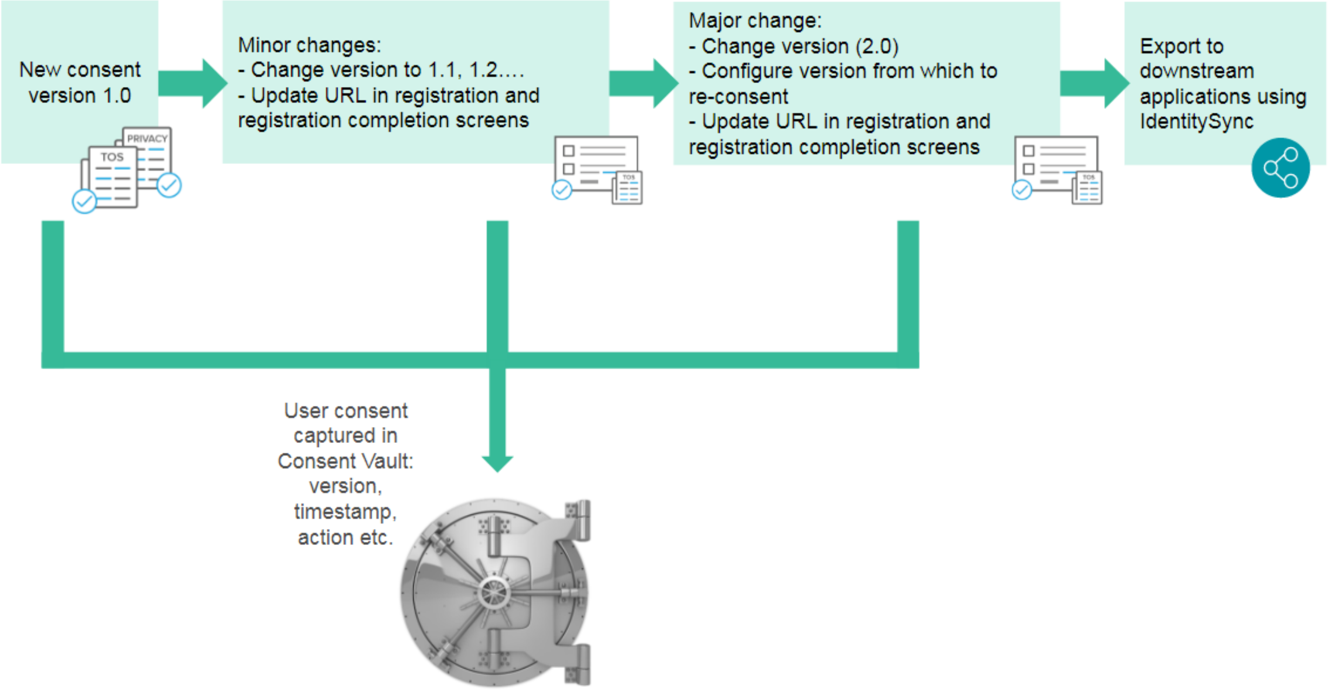
2. Create the consent statement, as explained above.
3. If any of your child sites require a separate consent statement, create them on the parent site at this stage.
4. The statement should be activated on the parent if it is the "main" statement for this group, i.e. required on the parent site and/or on most of the child sites. Activate the policy after the steps outlined above have been completed.

> If all the child sites require no additional consent statements, you have completed the configuration at this stage. If individual sites require separate statement activation, continue to the next steps.
>
> Remember that activating a statement on a child site breaks its inheritance from the parent.

5. In the site selector, select the child site for which you wish to apply the consent statement.
6. Go to the UI Builder and map the consent widget and checkbox to the relevant statement, on all the relevant screens (registration, registration completion, profile screens), as explained above.
7. Activate the statement for this child site.

Version Control

The following diagram illustrates the flow of version management and re-consent, from the perspective of a site admin:


New consent version 1.0
PRIVACY
TOS
Minor changes:
- Change version to 1.1, 1.2....
- Update URL in registration and registration completion screens
TOS
Major change:
- Change version (2.0)
- Configure version from which to re-consent
- Update URL in registration and registration completion screens
TOS
Export to downstream applications using IdentitySync
User consent captured in Consent Vault: version, timestamp, action etc.

Major vs. Minor Changes

The consent management tool gives you the option to update the version of the consent statement that is currently in effect, and the required date for consent renewal.

This means that when making a minor change to a statement, simply create a new version by updating the **Effective as of** field in the consent management dashboard. When the new version consists of major changes, that require existing users to re-consent, update both the **Effective as of** field (the active version number or date), and the **Re-consent cut-off** field. This means that users who consented to a version that is earlier than the number or date specifiied under **Re-consent cut-off**, will be required to re-consent (they will be presented a "Registration Completion" screen). Users who choose not to consent, will not be able to access the areas or features of your site that are available only to fully registered users.

In addition, you should configure the interval for validating that sessions of logged-in users are still valid, i.e. the consent version that is saved to their account is a valid one. If the consent version has changed, when the system performs session validation, that user will be presented with a 'Registration Completion' screen. If the user closes the screen without consenting, the user will be logged out (i.e., accounts.logout and socialize.logout are fired). Note that the configuration is different for mobile.

Implementation

Configure the interval for session validation:

1. Open the **Global Configuration** page in Gigya's Console.
2. Enter a new line with the **verifyLoginInterval** parameter.
3. Set a value in hours. Once every specified amount of hours, the system will check if all required schema fields have the relevant data, and that the session is in accordance with site policies. For example:

   ```
   verifyLoginInterval: 12
   ```

4. Save your changes.

Notes

- For more information, see Global Configuration.
- To ensure the validity of user sessions on mobile, you should set up some logic that calls accounts.verifyLogin periodically. This API validates the schema and policies of logged-in users, and logs them out if their session is invalid. You can also use this method in any case where you are not using Gigya's global configuration.

Change a version in the consent management dashboard:

1. Go to the Consent page of Gigya's Console.
2. Click the **Edit** button for the relevant consent statement.
3. Change the date or version number that the new version will go into effect (**Effective as of**).
4. If this is a major version change, also enter the date or version number for enforcing re-consent (**Re-consent cut-off**). This means, that users who consented to this agreement before this date or this version number, will be prompted to re-consent.

Update the URL in your screens, using the UI Builder:

1. Go to the Screen-Sets page of Gigya's Console.
2. Open the relevant screen in the UI Builder (e.g., registration).
3. Update the URL to the one that contains the updated terms of service, by using an HTML <a> tag. For example, if you have a checkbox with a label next to it that reads "I have read and understood the Terms of Service", where "Terms of Service" should link to the updated terms, the label property in the right-hand **Properties** pane should contain the link according to the following format: 'I have read and understood the <a class="terms-of-service" href="https://terms-of-service-v-2.0">Terms of Use</a>'.

- The active version is saved as part of the metadata of the Preferences Object. You can view the current version in the Consent page of Gigya's Console, or using the accounts.getSchema API call.
- There is no need to re-map any checkboxes or consent widgets that have already been mapped to this consent statement, as the version updates automatically.

As a result of a major version change:

- Users who have reconsented to an early version of this statement will be asked to reconsent (a **Registration Completion** screen will be displayed, containing the new consent version)
- When users re-consent, the action is captured and saved to the Consent Vault as a "Renewed" consent.
- New users consent to the active version, and that is saved to the Consent Vault as a "Granted" consent.

Ensuring Compliance Using IdentitySync

Sync consent-based user data to third-party applications, to support the following flows:

- Ensure that only data of users who have a valid consent statement associated with their profile is passed along.
- Handle data of users who have withdrawn their consent (e.g. archive, flag for deletion) and those whose consent has expired.

Consent enforcement is supported using IdentitySync, Gigya's ETL solution. When querying Gigya's database using the datasource.read.gigya.account component, use the **consent** parameter to retrieve only users of a given consent status. The possible values of this parameter are:

- valid - the users have consented to a given statement, and their consent is valid.
- expired - the users have consented to a given statement, but that consent is no longer valid, due to a version change to which they have not re-consented.
- notGranted - the users have either never consented to a given statement, or have withdrawn their consent.

Use as follows:

- To retrieve users with a valid consent, set up a job that queries Gigya's database, with the consent **status** parameter set to **valid**.
- To retrieve users whose consent has expired (for example, to trigger a re-consent flow in downstream applications), set up a job that queries Gigya's database, with the consent **status** parameter set to **expired**.
- To retrieve users who have never given consent, or who have withdrawn their consent, set up a job that queries Gigya's database, with the consent **status** parameter set to **notGranted**.

Click for a sample flow that exports to SFTP users with a valid consent status

```
{
 "name": "Export from Gigya to SFTP",
 "description": "account > rename > dsv > sftp",
 "steps": [
  {
   "id": "account",
   "type": "datasource.read.gigya.account",
   "params": {
    "select":
"profile.email,profile.lastName,profile.firstName,profile.gender",
//Extract these 4 data fields from Gigya
    "batchSize": 300,
    "from": "accounts",
    "deltaField": "lastUpdatedTimestamp",
    "maxConcurrency": 1,
    "consent":[
      {
        "id":"terms.tos",
        "status":"valid" //Select only the users for whom this statement
is valid;
      },
    ]
   },
   "next": [
    "rename"
   ]
  },
  {
   "id": "rename",
   "type": "field.rename", //Rename fields to match the format in the
target platform
   "params": {
    "fields": [
     {
      "sourceField": "profile.email",
      "targetField": "MAIL"
     },
     {
      "sourceField": "profile.lastName",
      "targetField": "NAME"
     },
     {
      "sourceField": "profile.firstName",
      "targetField": "FIRSTNAME"
     },
     {
      "sourceField": "profile.gender",
      "targetField": "GENDER"
     }
    ]
   },
    "next": [
    "dsv"
   ]
  },
  {
   "id": "dsv",
   "type": "file.format.dsv", //Create a DSV file with these users
   "params": {
    "fileName": "GIGYA_TO_SFTP_${now}.csv",
    "columnSeparator": ",",
    "quoteFields": true,
    "writeHeader": true,
    "lineEnd": "\n",
    "createEmptyFile": false
   },
   "next": [
    "sftp"
   ]
  },
  {
   "id": "sftp",
   "type": "datasource.write.sftp", //Write the file to SFTP
   "params": {
    "host": "...", //Insert SFTP credentials
    "username": "...",
    "password": "...",
    "remotePath": "GigyaDaily",
    "port": 22,
    }
  }
 ]
}
```

Click for a sample flow that exports to SFTP users with an expired consent status

```
{
 "name": "Export from Gigya to SFTP",
 "description": "account > rename > dsv > sftp",
 "steps": [
  {
   "id": "account",
   "type": "datasource.read.gigya.account",
   "params": {
    "select":
"profile.email,profile.lastName,profile.firstName,profile.gender",
//Extract these 4 data fields from Gigya
    "batchSize": 300,
    "from": "accounts",
    "deltaField": "lastUpdatedTimestamp",
    "maxConcurrency": 1,
    "consent":[
      {
        "id":"terms.tos",
        "status":"expired" //Select only the users for whom this
statement has expired;
      },
    ]
   },
   "next": [
    "rename"
   ]
  },
  {
   "id": "rename",
   "type": "field.rename", //Rename fields to match the format in the
target platform
   "params": {
    "fields": [
     {
      "sourceField": "profile.email",
      "targetField": "MAIL"
     },
     {
      "sourceField": "profile.lastName",
      "targetField": "NAME"
     },
     {
      "sourceField": "profile.firstName",
      "targetField": "FIRSTNAME"
     },
   {
    "sourceField": "profile.gender",
    "targetField": "GENDER"
   }
  ]
 },
  "next": [
  "dsv"
 ]
},
{
 "id": "dsv",
 "type": "file.format.dsv", //Create a DSV file with these users
 "params": {
  "fileName": "GIGYA_TO_SFTP_${now}.csv",
  "columnSeparator": ",",
  "quoteFields": true,
  "writeHeader": true,
  "lineEnd": "\n",
  "createEmptyFile": false
 },
 "next": [
  "sftp"
 ]
},
{
 "id": "sftp",
 "type": "datasource.write.sftp", //Write the file to SFTP
 "params": {
  "host": "...", //Insert SFTP credentials
  "username": "...",
  "password": "...",
  "remotePath": "GigyaDaily",
  "port": 22,
  }
  }
 ]
}
```

Click for a sample flow that exports to SFTP users who have not granted consent

```
{
 "name": "Export from Gigya to SFTP",
 "description": "account > rename > dsv > sftp",
 "steps": [
  {
   "id": "account",
   "type": "datasource.read.gigya.account",
   "params": {
    "select":
"profile.email,profile.lastName,profile.firstName,profile.gender",
//Extract these 4 data fields from Gigya
    "batchSize": 300,
    "from": "accounts",
    "deltaField": "lastUpdatedTimestamp",
    "maxConcurrency": 1,
    "consent":[
      {
        "id":"terms.tos",
        "status":"notGranted" //Select only the users who have never
granted, or have withdrawn, their consent;
      },
    ]
   },
   "next": [
    "rename"
   ]
  },
  {
   "id": "rename",
   "type": "field.rename", //Rename fields to match the format in the
target platform
   "params": {
    "fields": [
     {
      "sourceField": "profile.email",
      "targetField": "MAIL"
     },
     {
      "sourceField": "profile.lastName",
      "targetField": "NAME"
     },
     {
      "sourceField": "profile.firstName",
      "targetField": "FIRSTNAME"
     },
   {
    "sourceField": "profile.gender",
    "targetField": "GENDER"
   }
  ]
 },
  "next": [
  "dsv"
 ]
},
{
 "id": "dsv",
 "type": "file.format.dsv", //Create a DSV file with these users
 "params": {
  "fileName": "GIGYA_TO_SFTP_${now}.csv",
  "columnSeparator": ",",
  "quoteFields": true,
  "writeHeader": true,
  "lineEnd": "\n",
  "createEmptyFile": false
 },
 "next": [
  "sftp"
 ]
},
{
 "id": "sftp",
 "type": "datasource.write.sftp", //Write the file to SFTP
 "params": {
  "host": "...", //Insert SFTP credentials
  "username": "...",
  "password": "...",
  "remotePath": "GigyaDaily",
  "port": 22,
  }
  }
 ]
}
```

Additional Information

- Consent Vault
- Customer Consent
- Preference Center