1. Background

Yammer is integrating with Viva Topics. This effort aims to make community-sourced knowledge from Yammer easily accessible across Microsoft 365, establish a single concept of topics across Microsoft 365, enable premium Viva Topics experience in Yammer and create centralized management of topics for an organization. Viva Topics will bring together knowledge from across Microsoft 365 including SharePoint, Outlook, Yammer and so on.

As part of this, Yammer will adopt the Viva Topics platform for networks that are in Native Mode. While one can get premium Viva Topics experiences (Viva Topic pages, Viva Topic cards, etc.) with appropriate licensing, all existing topic capabilities within Yammer will not require any new licensing.

To achieve this, Yammer will need to make some changes to its Topic APIs. This document describes those changes along with recommendations to adapt.

2. Overview

Some API changes apply to all Yammer networks while others apply only to networks in Native Mode. This document details specific changes to Yammer’s V1 APIs. While V2 APIs are not documented publicly, a general commentary for changes to V2 APIs is included.

Upcoming changes will affect both V1 and V2 APIs. If you are reliant on undocumented V2 APIs, please ensure the changes outlined below are factored into your client.

3. Topic ID Structural Changes

3.1 ID type changes for all networks

- Topic IDs will now be represented as Strings instead of 64-bit Integers.
  - For example, a topic with ID 12345678910 (Integer) would become “12345678910” (String).

3.2 ID value changes for networks in Native Mode

- ID values are changing to represent Viva Topics and will be returned in responses.
  - Existing Yammer Topic IDs will continue be honored in API requests until further notice.
  - No guarantees of the form of the String are made. An example is YAM_aBcdefg_hij123KLmn4Opq.
  - These changes are highlighted in the examples below.

- Applying Topics should be done via ID rather than by name.
  - Applying topics by name over any API will no longer be supported.
  - You will now need to perform an additional step for attaching topics to new messages to support a more secure experience with Viva Topics. More details in section 6.7.

- Direct topic APIs, such as api/v1/threads/{id}/apply_topic, will require the topic ID to function.

- You can check whether a Yammer network is in Native Mode by using “m365_native_mode: [true|false]” on api/v1/networks/current.json.

- Note: During customer preview, reference data of the topic applied to the message may not be immediately available on posted messages. Topic data can be retrieved by requesting the message after posting. Messages applied to a topic can be retrieved with the “Messages about topic” API (see: documentation for /messages/about_topic/{id}.json ).
4. Preparing for Change
Some additional options are being made available to better support transitioning your clients to use Viva Topics, especially where semantics of the Topics are not directly compatible with legacy topics.

4.1 Intermediate ID Type Step
We recommend building your client generically to work with Topic IDs either as Strings or Integers. Prior to any of the ID type changes, an additional String value will be present for the IDs which will be transitioned as an optional aid. These will be of the form `{field_name}_str`. For example, on a topic model, `id` would be accompanied by `id_str`. This is a temporary value and will be removed after the ID type formally changes, at a date specified in broader communication.

Each API-specific section annotates the temporary value (highlighted in **bold cyan**). If your client code cannot support generic typing of the ID values, you should do the following:

- Update your client to use the “_str” suffix fields immediately, working with the String types returned.
- When the type change of the “id” field is released per the release timeline, update your client code to use the regular id fields again. At this stage, “id” and “id_str” will be identical.

If your client can support generic responses for the IDs being Integers or Strings, there is no need to use the intermediate step.

4.2 Early Access
Early access flags are being added for some APIs, for a limited time and for the sole purpose of allowing your clients to seamlessly transition to using Viva Topics.

Changes where this will be available are denoted with “**Early Access**” details in the API-specific changes.

5. API Reference Data Changes

5.1 V1 API Reference Data Changes

**Description of change:**

- In references[]: In all thread types (e.g. thread, shared_thread), Topic IDs in the topics field will be converted to String type (with the intermediate “_str” field to assist if you need it).
- In references[]: IDs under type: “topic” will be returned as Strings (again, with the intermediate “_str” available).

<table>
<thead>
<tr>
<th>Existing response</th>
<th>Updated response</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{ ... references:[ ... { type: “thread”, topics: [ { id: 262626, <strong>id_str</strong>: “262626”, type: &quot;topic&quot;} ]} }, { type: “topic”, id: 262626, ... references:[ ... { type: “thread”, topics: [ { id: “YAM_aBcdefg_hij123KLmn4Opq”, <strong>id_str</strong>: “YAM_aBcdefg_hij123KLmn4Opq”, type: &quot;topic&quot;} ]} }, { type: “topic”,</td>
<td></td>
</tr>
</tbody>
</table>
Changes in italic yellow indicate ID type and value changes.

Changes in bold cyan indicate temporary intermediate fields.

5.2 V2 API Reference Data

Reference data entities are already keyed by String typed IDs. Any assumptions made about the underlying data type cannot be guaranteed. For example, topics are keyed by the topic ID as a String, and it cannot be assumed that the String can be parsed as an Integer.

<table>
<thead>
<tr>
<th>Existing Model</th>
<th>Updated Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;reference_data&quot;: { &quot;topics&quot;: { &quot;262626&quot;: { &quot;id&quot;: 262626, &quot;id_str&quot;: &quot;262626&quot; } } }</td>
<td>&quot;reference_data&quot;: { &quot;topics&quot;: { &quot;YAM_aBcdefg_hij123KLmn4Opq&quot;: { &quot;id&quot;: &quot;YAM_aBcdefg_hij123KLmn4Opq&quot;, &quot;id_str&quot;: &quot;YAM_aBcdefg_hij123KLmn4Opq&quot; } } }</td>
</tr>
<tr>
<td>&quot;reference_data&quot;: { &quot;threads&quot;: [{ &quot;topics&quot;: [262626], &quot;topics_str&quot;: [&quot;262626&quot;] } }</td>
<td>&quot;reference_data&quot;: { &quot;threads&quot;: [{ &quot;topics&quot;: [&quot;YAM_aBcdefg_hij123KLmn4Opq&quot;], &quot;topics_str&quot;: [&quot;YAM_aBcdefg_hij123KLmn4Opq&quot;] } }</td>
</tr>
</tbody>
</table>

Changes in italic yellow indicate id type and value changes.

Changes in bold cyan indicate temporary intermediate fields.

6. Changes to API/V1 Endpoints

6.1 Get Topic by ID

**API Request:** GET /api/v1/topics/262626.json

**Description of change:**
- For All networks
  - ID will be returned as String.
- For Native Mode networks
  - ID will change.
  - Queries for the pre-existing Topic ID ("legacy" ID) will be honored, but the new Viva Topic ID will be returned as the "id" in the response. In this example, querying for Topic 262626 would return the topic YAM_aBcdefg_hij123KLmn4Opq. The legacy ID 262626 may be used in client requests, but we recommend updating to use the new IDs as they are discovered.
In this case, a request for GET /api/v1/topics/YAM_aBcdefg_hij123KLmn4Opq.json would yield the same result as for topic 262626.

- Viva Topics will not have normalized_name or permalink. These fields will remain present but always be empty.
- To support message posting with Topics, which requires extra security, topics should be authorized for being applied to new conversations. To generate a token for this, include query parameter preauthorize_topics=true in your request. This will be returned as “token” in the topic response.
  - You can use this token only to apply this topic to messages that this user creates. This is neither an authentication token nor will it be honored by any other Viva product or API.
  - This token is specific to the requesting user and the requested topic.
  - This token has a 1-hour validity, after which a new token will need to be requested.
  - The token is a standard JWT token; the expiry time can be inspected in the token’s claims.
  - Modifying any part of the token will render it invalid.
- Early Access: Topic authorization tokens can be requested even for non-native mode networks. You can request these tokens in non-native mode networks but is not necessary for legacy topic interactions.

<table>
<thead>
<tr>
<th>Existing response</th>
<th>Updated response</th>
</tr>
</thead>
</table>

Changes in italic yellow indicate id type and value changes.
Changes in bold cyan indicate temporary intermediate fields.

6.2 Messages about Topic

API Request: GET api/v1/messages/about_topic/262626.json
Public documentation: https://developer.yammer.com/docs/messagesabout_topicidjson

Description of change:
- V1 Reference Data changes as described in the V1 API Reference Data section.

6.3 Received Messages

API Request: /api/v1/messages/received.json
Public documentation: https://developer.yammer.com/docs/messagesreceivedjson

Description of change:
- V1 Reference Data changes as described in the V1 API Reference Data section.
6.4 Subscriptions to Topics (“Following”)

**API Request 1:** GET /subscriptions/to_topic/262626.json
**API Request 2:** GET /subscriptions/12345.json
**Public documentation:** [https://developer.yammer.com/docs/subscriptionsto_topicidjson](https://developer.yammer.com/docs/subscriptionsto_topicidjson)

**Description of change:**
- All networks:
  - `target_id` will be returned as `String`.
- Native mode networks:
  - `target_id` will change to reflect the Viva Topic ID.
  - Queries for the “legacy”/existing `id` will be honored, but the updated `id` will be returned as the “`id`” in the response. In this example, querying for topic 262626 would return the topic `YAM_aBcdefg_hij123KLmn4Opq`. The legacy id 262626 may be used in requests, but we recommend updating to use the new IDs.

<table>
<thead>
<tr>
<th>Existing response</th>
<th>Updated response</th>
</tr>
</thead>
<tbody>
<tr>
<td>```json</td>
<td></td>
</tr>
</tbody>
</table>
| {
|   type: "subscription",
|   url: ".../api/v1/subscriptions/12345",
|   target_id: 262626,
|   target_id_str: "262626",
|   target_url: ".../api/v1/topics/262626",
|   target_web_url: ".../topics/262626",
|   target_type: "topic",
|   id: 12345
| ``` | ```json |
| ```json |
| { 
|   type: "subscription",
|   url: ".../api/v1/subscriptions/12345",
|   target_id: "YAM_aBcdefg_hij123KLmn4Opq",
|   target_id_str: "YAM_aBcdefg_hij123KLmn4Opq",
|   target_url: ".../api/v1/topics/YAM_aBcdefg_hij123KLmn4Opq",
|   target_web_url: ".../topics/YAM_aBcdefg_hij123KLmn4Opq",
|   target_type: "topic",
|   id: 12345
| ``` |

*Changes in *italic yellow* indicate id type and value changes.*

*Changes in *bold cyan* indicate temporary intermediate fields.*

6.5 Adding Subscriptions

**API Request:** POST api/v1/subscriptions
**Public documentation:** [https://developer.yammer.com/docs/subscriptions](https://developer.yammer.com/docs/subscriptions)

**Description of change:**
- All Networks:
  - The supported type for `target_id` in the request body is changing to `String`.
    - IDs passed as integer will continue to be honored.

6.6 Removing Subscriptions

**API Request:** DELETE api/v1/subscriptions
**Public documentation:** [https://developer.yammer.com/docs/subscriptions-1](https://developer.yammer.com/docs/subscriptions-1)

**Description of change:**
- All Networks:
  - The supported type for `target_id` in the request body is changing to `String`.
    - IDs passed as integer will continue to be honored.

6.7 Attaching Topics to a New Message

**API Request:** POST api/v1/messages
**Public documentation:** [https://developer.yammer.com/docs/messages-json-post](https://developer.yammer.com/docs/messages-json-post)

**Description of change:**
- All Networks
Topic IDs in the request body are supported as Strings.
- For “legacy” Topic IDs, passing as Integers will be honored.
- Topics by ID are represented by topic_id1={ID} .. topic_id20={ID}
- Topics in the response are updated in line with the changes in the ID type changes for all section.

- Native Mode networks:
  - Applying topics by name will no longer be supported.
  - Viva Topics can be ambiguously named, with Topics potentially sharing the same name with other topics with other meanings or purposes.
  - Topics will not be implicitly created.
  - Topics require an additional layer of security during message posting. Topics should be attached using a secure token.
    - The secure topic token can be acquired from api/v1/topics/{id}.
    - More detail on requesting this token is in section 6.1.
  - Requests made to apply topics should include Topic topic_authorization_token, enumerated as with the existing API.
    - Eg: topic_authorization_token1={token 1}
    - topic_authorization_token20={token 20}
    - As with existing APIs, topic tokens beyond the 20th will be truncated.
  - Message posting will not honor Topic IDs passed in the request payload; this may be added in a future update.
    - Tip: Applying a topic after a message is created can be done in a single subsequent request and is equally secure; applying topics via this API may be an easier option for your solution. See section 6.8.
  - Early Access: Support will be added for all networks to submit topic_authorization_tokens, though it is less performant and not required for non-native mode clients. If any combination of Tokens IDs and Names corresponding to the same topic are provided, that topic will only be applied once. See section 6.1 for early access of token generation.

<table>
<thead>
<tr>
<th>Existing request (Non-Native Mode)</th>
<th>Updated request (Non-Native Mode)</th>
</tr>
</thead>
</table>

Changes in italic yellow indicate id type and value changes.

<table>
<thead>
<tr>
<th>Existing request (Native Mode)</th>
<th>Updated request (Native Mode)</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>{ body: ..., group_id: 1234, network_id: 107, topic_id1: 262626, topic_id2: 272727 }</code></td>
<td><code>{ body: ..., group_id: 1234, network_id: 107, topic_authorization_token1: &quot;eyJh...kb28&quot;, topic_authorization_token2: &quot;eyJh...rBkw&quot; }</code></td>
</tr>
</tbody>
</table>

Changes in italic yellow indicate id type and value changes.
6.8 Applying Topics to Existing Threads

**API Request:** POST api/v1/threads/{thread_id}/apply_topic

**Public documentation:** N/A

**Description of change:**
You can apply topics to conversations using Topic IDs.

- **All Networks:**
  - Applying topics by ID is supported.

- **Native Mode Networks:**
  - Applying topics by name is not supported.
  - Applying topics by topic authorization token (as in sections 6.1 & 6.7) is not necessary or supported.

<table>
<thead>
<tr>
<th>Existing request</th>
<th>Updated request</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{</td>
<td>`{</td>
</tr>
<tr>
<td>topic: Secret+Sauce</td>
<td>topic_id: “262626”</td>
</tr>
<tr>
<td>}</td>
<td>}</td>
</tr>
</tbody>
</table>

*Changes in italic yellow indicate id type and value changes.*

6.9 My Feed

**API Request:** POST api/v1/messages/my_feed.json

**Public documentation:** [https://developer.yammer.com/docs/messagesmy_feedjson](https://developer.yammer.com/docs/messagesmy_feedjson)

**Description of change:**
- Topic references, as outlined in the V1 API Reference Data section.
7. Changes to API/V2 Endpoints

7.1 V2 API - Topics in feed and thread responses

While Yammer’s v2 API endpoints are not formally supported, the guidance for Topic IDs applies generally here too.

Topic ids on models are changing to strings, same as V1 APIs.

<table>
<thead>
<tr>
<th>Existing Model</th>
<th>Updated Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;reference_data&quot;: {</td>
<td>&quot;reference_data&quot;: {</td>
</tr>
<tr>
<td>&quot;topics&quot;: {</td>
<td>&quot;topics&quot;: {</td>
</tr>
<tr>
<td>&quot;262626&quot;: {</td>
<td>&quot;id&quot;: &quot;YAM_abcdEfg_hij123Klmn4Opq&quot;,</td>
</tr>
<tr>
<td>&quot;id&quot;: &quot;262626&quot;,</td>
<td>&quot;id_str&quot;: &quot;YAM_abcdEfg_hij123Klmn4Opq&quot;,</td>
</tr>
<tr>
<td>&quot;id_str&quot;: &quot;262626&quot;</td>
<td>&quot;id&quot;: &quot;YAM_abcdEfg_hij123Klmn4Opq&quot;,</td>
</tr>
<tr>
<td>}</td>
<td>&quot;id_str&quot;: &quot;YAM_abcdEfg_hij123Klmn4Opq&quot;</td>
</tr>
<tr>
<td>}</td>
<td>}</td>
</tr>
</tbody>
</table>

Changes in **italic yellow** indicate id type and value changes.
Changes in **bold cyan** indicate temporary intermediate fields.

8. Changes to Data Export

8.1 Data Export

To help clients understand the source of the topics, a new csv file, Viva_Topics.csv, is being included in Yammer’s Data Export. Topics.csv will not contain Viva Topic references. To best support existing data export flows, Viva_Topics.csv will start with the same columns as Topics.csv. Any new data will be made available in subsequent columns.

Topics.csv contains these fields today:

- ID
- NAME
- API_URL
- CREATED_BY
- CREATED_AT
- DESCRIPTION

Viva_Topics.csv will contain these fields:

- ID
- NAME
- API_URL
- CREATED_BY
- CREATED_AT
- DESCRIPTION
- LEGACY_ID
- MERGED_IDS

Note:

- While Viva Topics are in Customer Preview, NAME and DESCRIPTION will be blank, and data must be retrieved via the API listed in API_URL.
- CREATED_BY will never be present, as Yammer no longer tracks this. This information is in Viva Topics' systems and will need to be retrieved using Viva Topics.
• **LEGACY_ID** will contain the Yammer Topic ID as it used to appear in Topics.csv. Other topics from other sources, such as user-created Topic Pages in Viva, will not have a value present.
• **MERGED_IDS** represents a list of topics that have been merged into this topic.

Viva_Topics.csv will only contain data for Viva Topics used in Yammer. This includes (1) any Yammer topics imported as Viva Topics and (2) any Viva Topics which have been used in Yammer. As a result, you cannot assume that the Yammer data export will provide an exhaustive list of all available Viva Topics.

While a network is being transitioned to Viva Topics, data will exist for both Viva Topics and Legacy Topics. After a network has been transitioned, Topics.csv will no longer contain any data.