Social IT

Live Feed

PDF generated using the open source mwlib toolkit. See http://code.pediapress.com/ for more information.
PDF generated at: Wed, 29 Nov 2017 20:25:25 PST

Introduction

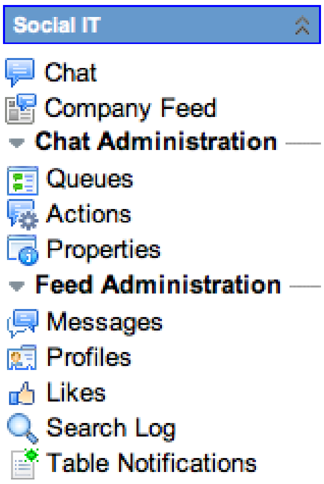
Social IT

Note: *This article applies to Fuji and earlier releases. For more current information, see Legacy: Chat* [1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.**

Overview

Social IT is the use of social media technology (such as instant messaging and microblogging) to enhance IT operations. The benefits of Social IT include:

- Improves communication throughout an organization.
- Uses familiar tools from everyday life; users do not need to learn complicated, new products.
- Provides a quick way for business users to contact the service desk.
- Empowers IT and non-IT employees alike to answer IT-related questions. Users get answers faster, and IT staff have more time to move the business forward with technology.
- Integrates task data with conversation history for quicker resolutions and improved knowledge management.



Social IT application

Several Social IT features are available in a ServiceNow instance:

- **Chat**
  - Chat provides real-time communication between users via instant messaging.
  - Chats may be one-to-one or between multiple users (chat rooms).
  - Chat rooms can be linked to task records. Users can work together to solve issues, and conversation history can be shared by everyone who needs to reference it.
- **Help Desk Chat**
  - End users can communicate directly with service desk staff via instant messaging from the Employee Self-Service Portal.
  - Service desk staff can resolve basic issues in real time or create incidents directly from chat requests for more extensive issues.
- **Live Feed**: Users can view messages from other users or about specific business services. This information forms a searchable knowledge source for sharing information within an organization.

Getting Started with Social IT

To get started with Social IT in ServiceNow:

1. **Establish use guidelines.** Social media can improve communication and aid productivity. To get the most out of these tools, establish clear and simple social media guidelines that foster information sharing and a comfortable work environment.
2. **Activate the plugins.**
   - Chat plugin: activates the Chat and Help Desk Chat features.
   - Live Feed plugin: activates the Live Feed feature.
3. **Configure security settings.** Users must log in to use the Social IT features in ServiceNow. The standard system security settings and Social IT-specific security settings are available. To learn more about setting up these features, refer to the plugin activation pages.
   - Chat rooms may be *Public* (any user can join) or *Private* (only invited users can join).
   - Administrators can limit who can read chat messages on tasks and who can create chat rooms.
   - Live Feed is domain-separated (users can only read messages posted in their domain).

References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/use/using-social-it/concept/c_Chat.html

Live Feed

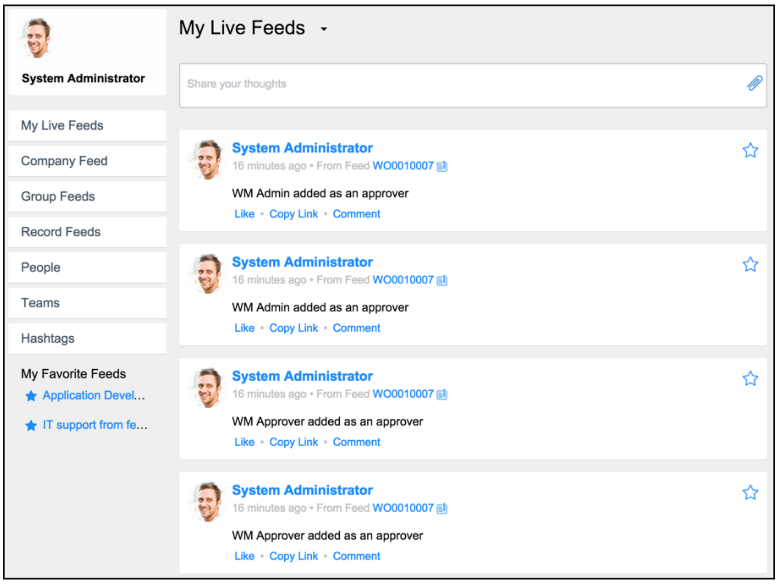
Note: *This article applies to Fuji and earlier releases. For more current information, see Live Feed* [1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.**

Overview

Live feed [2] is a social IT application that provides a place to post and share content in a ServiceNow instance. This content forms a searchable knowledge source for sharing information within an organization. Main features include the following:

- Users can post, reply to, and rate messages, including links and images.
- Administrators can set up automatic messages that are generated when specific records are updated throughout the system.
- Users can subscribe to receive email notifications when new messages are posted.

Live feed includes different types of feeds [3].



An example of the My Live Feeds feature

See Live Feed UI Overview for general instructions on using live feed features. See Using Live Feed to Work on Records for information on how to interact with live feed on record forms.

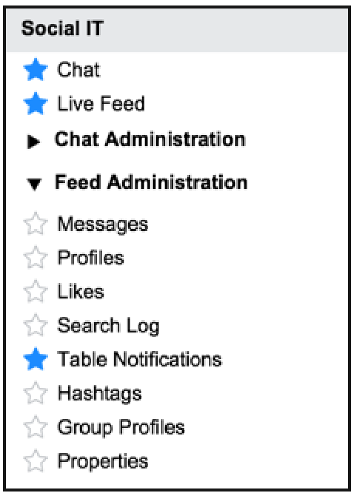
Note: *Many of these features are part of live feed v2, which is available starting with the Fuji release. Live Feed v2 is active for all new instances by default. If you are upgrading from an earlier version of ServiceNow, you need to activate live feed v2 to use these features. If you are using live feed v1, or have not activated live feed v2, see Legacy: Live Feed.*

Browser Support

The Live Feed v2 plugin does not support Internet Explorer 7 through 9. Users who access the instance from those browser versions can use Live Feed v1 functionality. Users who access the instance from browsers that are compatible with HTML5 can use Live Feed v2 functionality.

Menus and Modules

The **Social IT** application appears in the application navigator, with modules for using and administering live feed.



Social IT application

For all users:

- **Live Feed:** opens the live feed application for the current user.

Administrators and users with the live_feed_admin role see the following items under **Feed Administration**:

- **Messages**, **Profiles**, **Likes**, **Search Log**, **Hashtags**, and **Group Profiles:** provide access to tables for viewing live feed data (see New Tables). Extending or modifying data in these tables is not recommended.
- **Table Notifications:** opens the Live Table Notifications table, where administrators can define automatically generated messages. To learn more, see Setting Up Live Feed Table Notifications.
- **Properties:** opens the Social IT Properties page.

Activating Live Feed

Live feed is active for all instances. Live feed v2 is active on all new instances starting with the Fuji release. For instances upgrading from a version prior to Fuji, an administrator can activate the Live Feed v2 plugin.

Click the plus to expand instructions for activating a plugin.

If you have the admin role, use the following steps to activate the plugin.

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.

   If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. [Optional] If available, select the **Load demo data** check box.

   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click **Activate**.

Enhancements

Fuji

- Table notification filter conditions support changes, changesTo, and changesFrom operators.
- Knowledge article IDs added to live feed messages are automatically converted to links to a knowledge base in organization's instance.
- URLs added to live feed messages are automatically converted to links.
- Code added to live feed messages retains newline and tab formatting.
- Using My Feed, new messages and replies to existing messages can be sent to all users in the feed.
- Using Company Feed, replies to existing messages can be sent to all users in the feed.
- A new property, `glide.live_feed.auto_join_document_group`, enables automatic membership to a record feed whenever the associated document is visited and the user clicks Show Live Feed. This behavior is also active if the `glide.ui.show_live_feed_activity` property is set to true.

Eureka

- Domain separation is enabled for the Live Feed application. This enhancement adds a **sys_domain** field to live feed tables. For more information, see Domain Assignment.

References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/use/live-feed/concept/c_GetStartedWithLiveFeed.html
[2] https://docs.servicenow.com/bundle/helsinki-servicenow-platform/page/use/live-feed/concept/c_GetStartedWithLiveFeed.html
[3] https://docs.servicenow.com/bundle/helsinki-servicenow-platform/page/use/live-feed/reference/r_LiveFeedUIOverview.html

Use

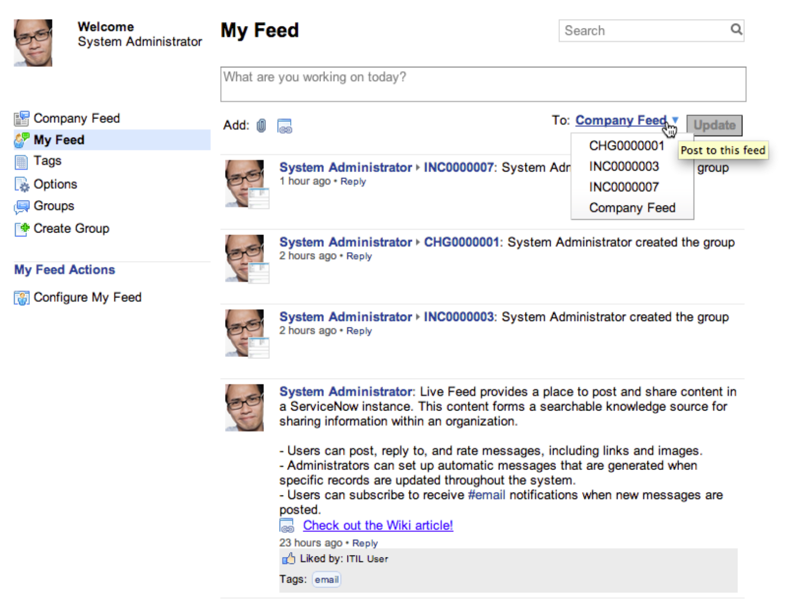
Using Live Feed

Overview

Live feed is a social IT application that provides a place to post and share content in a ServiceNow instance. This content forms a searchable knowledge source for sharing information within an organization. Main features include the following:

- Users can post, reply to, and rate messages, including links and images.
- Administrators can set up automatic messages that are generated when specific records are updated throughout the system.
- Users can subscribe to receive email notifications when new messages are posted.

Live feed includes different types of feeds, including **My Feed**, which shows all messages that are relevant to a specific user. See Using Live Feed for instructions on how to use all live feed features. See Using Live Feed to Work on Records for information on how to interact with live feed on record forms.

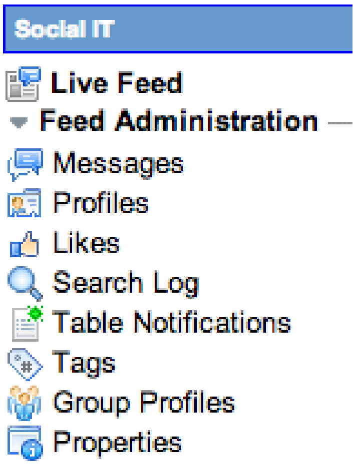


An example of the My Feed feature in live feed

Note: *The live feed functionality changed significantly in live feed v2 (starting with the Fuji release). This page describes live feed v1. For information on live feed v2, see Live Feed.*

Application and Modules

The **Social IT** application appears in the application navigator, with modules for using and administering live feed.



Social IT application

For all users:

- **Live Feed:** opens the live feed application for the current user. To learn more, see Using Live Feed.

Administrators and users with the live_feed_admin role see the following items under **Feed Administration**:

- **Messages**, **Profiles**, **Likes**, **Search Log**, **Tags**, and **Group Profiles:** provide access to tables for viewing live feed data (see New Tables). Extending or modifying data in these tables is not recommended.
- **Table Notifications:** opens the Live Table Notifications table, where administrators can define automatically generated messages. To learn more, see Setting Up Live Feed Table Notifications.
- **Properties:** opens the Social IT Properties page.

Activating the Plugin

Before activating this plugin, consider the installed components, dependencies, and impact.

- **Installed Components:** Include an application, tables, business rules, a script include, events, email notifications, a script action, a user role, and security rules. For more details, see Installed Components.
- **Dependencies** (installed automatically): Social IT Infrastructure.
- **Impact:** The plugin installs new features; it does not overwrite or impact current configurations.

This plugin is automatically activated for new instances. For older instances, activate the plugin.

Instructions for activating a plugin.

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.
3. [Optional] Select the **Load demo data** check box.

   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when first activating the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.

   If the plugin depends on other plugins, these plugins and their activation status are listed.
4. Click **Activate**.

Installed Components

New Tables

The following tables are added:

Display Name (Table Name)	Description
Live Feed Searches (live_feed_search)	Stores live feed text searches. Access at **Social IT > Feed Administration > Search Log** (administrator, text_search_admin, and live_feed_admin).
Live Tag Follows (live_tag_follow)	Maintains users' follow preferences for tags.
Live Follows (live_follow)	Maintains users' follow preferences for user feeds.
Live Group Profile (live_group_profile)	Stores group properties, including name, description, and public status.
Live Group Member (live_group_member)	Maintains the member lists for groups.
Link attachments (live_link)	Stores links and attachments that are added to messages (read access for administrator and live_feed_admin).
Messages Liked by (live_message_like)	Maintains *like* ratings for posts. Access at **Social IT > Feed Administration > Likes** (administrator, live_feed_admin, and chat_admin).
Live Message Tag (live_message_tag)	Stores messages associated with user-created tags (read access for administrator and live_feed_admin). Appears as a related list on Live Tag records.
Live Messages (live_message)	Stores messages. References sys_user table. Access at **Social IT > Feed Administration > Messages** (administrator and live_feed_admin).
Live Profiles (live_profile)	Stores user live feed profiles. References sys_user table. Access at **Social IT > Feed Administration > Profiles** (administrator and live_feed_admin).
Live Table Notification (live_table_notification)	Generates automatic table notifications, such as when a user reads live feed for the first time and when a high priority incident is opened. Access at **Social IT > Feed Administration > Table Notifications** (administrator, chat_admin, and live_feed_admin).
Live Tag (live_tag)	Maintains user-created tags. Access at **Social IT > Feed Administration > Tags** (administrator and live_feed_admin).

Business Rules

The following business rules are added:

- **Feed Group Creator Becomes Admin:** designates the group creator as the group administrator.
- **live feed events:** runs on task insert, update, and delete. Triggers event associated with the **Live Feed Update** Script action that processes Live Table Notifications to auto-generate live feed messages.
- **Live feed group member update events:** generates a notification event when member state changes (invited, accepted, declined, left, rejected, request, request_accepted).
- **Live feed new group member events:** generates a notification event when new members are added.
- **live feed profile events:** runs on live_profile insert/update/delete, triggers event associated with the **Live Feed Update** script action that processes Live Table Notifications to auto-generate live feed messages.
- **Live message events:** runs on live_message, notification event trigger for new live messages.
- **Live message like events:** runs on live_message_like, notification event trigger for new like records.
- **LiveFeed Group Member Visibility:** ensures users can only see the members list for public groups and groups they belong to.
- **LiveFeed Group Profile Validation:** ensures that a public group is visible.

- **LiveFeed Group Profile Visibility:** ensures that the list of all groups only displays public groups, private groups that are visible, and groups the user belongs to.
- **LiveFeed Membership Changes:** enforces that only the group administrator and users with live_feed_admin role can manage membership for a group.
- **LiveFeed Single Group Membership:** ensures that a user is not added multiple times to the same group.

Script Includes

The following script includes are added:

- **LiveMsgUtil:** helper functions for working with live_message records.
- **LiveFeedFilter:** query filters for working with live_feed records.
- **LiveFeedUtil:** helper functions for working with live_feed records.

Events

The following events are registered:

- `live_message_like.inserted`: triggers an email notification when another user *likes* a message that a user posts.
- `live_message.replied`: triggers an email notification to all users in a conversation thread when a user posts a reply.
- `live_message.inserted`: triggers an email notification to subscribed users when a new message is posted.
- `live_feed.update`: triggers the Live Feed Update script action.
- `live_message.group_inserted`: triggers an email notification to subscribed users when a new message is posted to a group.
- `live_group_member.invited`: triggers an email notification when a user is invited to join a private group.
- `live_group_member.accepted`: triggers an email notification when a user accepts an invitation to join a group.
- `live_group_member.declined`: triggers an email notification when a user declines an invitation to join a group.
- `live_group_member.request`: triggers an email notification when a user requests to join a private group.
- `live_group_member.request_accepted`: triggers an email notification when a user's request to join a private group is accepted.
- `live_group_member.rejected`: triggers an email notification when a user's request to join a private group is rejected.
- `live_group_member.left`: triggers an email notification when a user leaves a group.

Email Notifications

The following email notifications are added:

- **Live Message Liked:** sends an email to the creator of a message when another user *likes* the message (*live_message_like.inserted* event).
- **Live Message Reply:** sends an email to all users in a conversation thread when a user posts a reply (*live_message.replied* event).
- **Live Message All Subscription:** sends an email to subscribed users when any message (new or reply) is posted.
- **Live Message New Posts Subscription:** sends an email to subscribed users when a new (not reply) message is posted.
- **Live Group Member Accepted:** sends an email when a user accepts an invitation to join a group.
- **Live Group Member Declined:** sends an email when a user declines an invitation to join a group.

- **Live Group Member Invited:** sends an email when a user is invited to join a group.
- **Live Group Member Left:** sends an email when a user leaves a group.
- **Live Group Member Request:** sends an email when a user requests to join a private group.
- **Live Group Member Request Accepted:** sends an email when a user's request to join a private group is accepted.
- **Live Group Member Request Rejected:** sends an email when a user's request to join a private group is rejected.
- **Live Message Group Subscription:** sends an email to subscribed users when a message is posted to a group.
- **Live Message Group Reply Subscription:** sends an email when a member replies to a message in a group.

Script Actions

The following script action is added:

- **Live Feed Update:** generates system messages when an update occurs on a table that is relevant to live feed.

User Roles

The following user roles are added:

Role	Description
live_feed_admin	Can manage live feed functions.
chat_admin	Can manage chat functions (if the Chat plugin is activated).

Security Rules (ACL) on Existing Tables

The following security rules (ACL) on existing tables are included:

Table	Description
sys_attachment	Allow users to read and write live feed and live profile attachments

Table Notifications

The following live feed table notifications are included:

- **User joined live feed:** *Active* by default.
- **High priority incident opened:** Not active by default.
- **Incident resolved:** Not active by default.

Properties

To access live feed properties, navigate to **Social IT > Feed Administration > Properties**.

The following properties are added:

- `glide.live_feed.max_popular_tags`: the maximum number of popular tags displayed.
- `glide.live_feed.max_recent_tags`: the number of recent tags to show in the tag display.
- `glide.live_feed.my_feed_enabled`: enables user access to My Feed capabilities.
- `glide.live_feed.company_feed_exclude_groups`: prevents messages that belong to a group from appearing on the Company Feed.

  To configure this property (Calgary release): navigate to the **Social IT > Feed Administration > Properties** page and modify the **Do not show messages that belong to a group on the Company Feed** option.
- `glide.live_feed.message_poll_interval`: how frequently live feed is updated with new messages.

To configure this property (Calgary release): navigate to **Social IT > Feed Administration > Properties**. Enter the polling value in the **Live message poll interval time in ms.** field.

- `glide.ui.show_live_feed_activity`: displays two tabs on a record form's activity formatter: **Live Feed**, which allows users to collaborate on the document feed, and **Activity**, which shows a summary of all activity for the record (Calgary Release). This property is not enabled by default.

Scripted Web Services

This following scripted web service is installed:

- **PostToLiveFeed:** allows external systems to post to live feed. This service is inactive by default. Optional input parameters allow you to:
  - Define the user or profile to use; otherwise the message posts from the authenticated soap user.
  - Define the group (using the groupid); otherwise the message posts to the company feed.
  - Include a link in the post.

Sample SOAP envelope

```xml
<SOAP-ENV:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:m="http://www.service-now.com" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tns="http://www.service-now.com/PostToLiveFeed"  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">

 <SOAP-ENV:Body>

  <m:execute>

   <message xsi:type="xsd:string">Post this message to the Live Feed</message>

   <!-- optional sys_id of a sys_user record to use to get a live profile -->

   <user xsi:type="xsd:string"></user>

   <!-- optional sys_id of a live profile to post as -->

   <profile xsi:type="xsd:string"></profile>

   <!-- optional sys_id of a live group to post the message in-->

   <group xsi:type="xsd:string"></group>

   <!-- optional URL of a link to include in the message-->

   <linkurl xsi:type="xsd:string"></linkurl>

   <!-- optional name for the linkurl parameter-->

   <linkname xsi:type="xsd:string"></linkname>

  </m:execute>

 </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

User Preferences

Live feed uses the following user preferences:

- **live_tag.feed.order:** tracks user's choice of tag display sorting.
- **live_message.feed.last_display:** tracks last feed user has viewed. The last viewed feed is displayed the next time the user opens live feed. The first time a user opens live feed, the Company Feed is displayed.

Security

Live Feed Table Access Rules

By default, administrators and users with the live_feed_admin role can view all live feed tables and can modify the following live feed tables:

- **Live Table Notification [live_table_notification]:** set up automatic messages that are generated when specific records are updated throughout the system (see Setting Up Live Feed Table Notifications).
- **Live Feed Message [live_message]:** can be modified if necessary (such as to delete an inappropriate message or restore a message that a user accidentally deleted), but should almost always be managed by system functionality.

Extending or modifying data in any live feed chat table is **not** recommended.

Domain Separation

Using domain separation with the Live Feed application allows administrators to keep user content within a specific domain. Parent and child domains define the live feed content that can be viewed and shared with other users.

- A user in a parent domain can see users and content within their domain and within all child domains that are lower in the domain hierarchy.
- A user in a child domain can see users and content within their domain but cannot see the parent domain or other child domains at the same level in the domain hierarchy.

For more information about parent-child relationships, see Data Separation.

Enabling Domain Separation for Live Feed

Domain separation for the Live Feed application is available starting with the Eureka release. For current users upgrading to Eureka, a **sys_domain** column is added to existing live feed-related database tables. For existing database records, the value of the **sys_domain** field is set to empty (global). This allows the new domain separation feature to work with existing live feed records.

For more information, see Domain Separation.

Showing or Hiding the Live Feed Application for a Domain

The administrator can show or hide live feed for a domain. This includes:

- Displaying live feed in the application navigator.
- Following a document feed for a record.
- Viewing live feed from a record.

For more information, see Enabling and Disabling Application Menus or Modules.

Using Live Feed with Domain Separation

All of the live feed features are available to users within a domain and work the same way, with some exceptions to visibility.

- **Posting and sharing content**
  - Users in a child domain can interact with other users in the same domain. This includes posting and replying to messages, deleting messages, attaching files and links, and rating content.
  - Users in a parent domain can interact with other users in the same domain and with users in any child domains. In addition to the above interactions, users in a parent domain can see and add messages to threads in any child domain. Messages that are added to child threads by parent users are given the visibility of the child domain, not the parent domain.
- **Viewing content**
  - Users in a child domain can view content that resides within their domain. This includes filtering by feed, sorting, searching, and viewing older messages.
  - Users in a parent domain can view content that resides withinin their domain and within any child domains.

- Using My Feed shows messages, groups, and tags based on the user's domain. Showing another user's feed only shows posts visible in the current user's domain.
- **Using tags**
  - Tags are separated by domain. Users in a child domain can use all of the tag functions within their domain. This includes tagging messages, viewing available tags, changing tag names, and searching and filtering by tag.
  - Users in a parent domain can use all of the tag functions within their domain and can see tags in any child domains.
- **Using groups**
  - Groups are separated across domains. Users in a child domain can join and follow groups within their domain, and can invite other users within their domain to join groups. Users in a parent domain can join and follow groups that belong to their domain or to any child domains.
- **Exceptions to groups**
  - New groups are created in the user's domain. However, if the group has a record associated with it, then the group is created in record's domain.
  - New threads in an existing group are created in the domain of the existing group.
  - When a user is creating a new group from a document, if the document domain is not empty or global, set the group domain to be the document's domain.
  - For a document-generated group, when the domain of the document changes, the domain of all related live feed records changes as well.
- **Subscribing to email notifications**
  - Users can subscribe to email notifications from users and groups only within their domain.

Limiting Live Feed Access by Role

All active users in the instance have access to live feed by default. To limit access by role:

1. Define the roles that allow users to see the Live Feed module.
   1. Right-click the icon next to the Live Feed module and select **Edit Module**.
   2. Enter the roles that have access in the **Roles** field.
   3. Click **Update**.
2. Define the roles that allow users to see live feed from a mobile device.
   1. Navigate to **System Definition > Applications (Mobile)**.
   2. Click **Live Feed**.
   3. Enter the roles that have access in the **Roles** field.
   4. Click **Update**.
3. Create access control rules to limit who can view the live_feed pages.
   - **Type:** *ui page*
   - **Operation:** *read*
   - **Name:** create one access control rule record for each of the following pages
     - live_feed
     - live_feed_small
   - Under the **Requires role** related list, add roles to define who can access the live feed pages. Users who do not have these roles cannot access live feed.

Message Content

For security reasons, any HTML code is automatically stripped out of a message before it is posted. This measure ensures that users cannot modify any page settings by posting a message.

Users can remove their own messages from feeds. If necessary, administrators can remove inappropriate messages that are posted by any user.

To remove a message from all feeds:

1. Navigate to **Social IT > Feed Administration > Messages**.
2. Open the message to be removed.
3. In the **State** field, select **deleted**. You may need to configure the form to add this field.
4. Click **Update**.

Group Security

Administrators can restrict who can create groups by modifying the following access control rule:

- **Table:** *Live Group Profile [live_group_profile]*
- **Operation:** *create*

Restricting Tag Renaming

Administrators can restrict who can rename tags by modifying the following access control rule:

- **Table:** *Live Tag [live_tag]*
- **Operation:** *write*

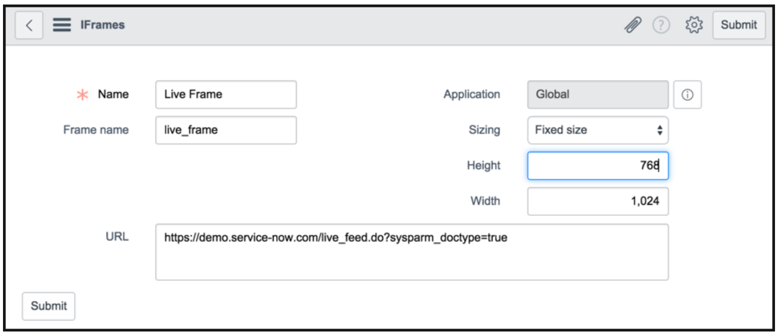
Providing Access to Live Feed from Content Management Pages

You can provide access to Live feed from pages built in the Content Management System (CMS). For example, allow an end user to access your company feed via the ESS portal.

The ESS Portal template includes the **Portal - Live page** and **Live Feed** dynamic block (requires the Live feed plugin). To provide access to Live feed from CMS pages, add the Live Feed dynamic block to a CMS page or include the **Portal - Live page** in a site.

To add Live feed in an iFrame:

1. Navigate to **Content Management > iFrames**.
2. Click **New**.
3. Enter the iFrame block details:
   - **Name**: enter a unique name (**Live Frame**)
   - **Frame Name**: enter a frame name (**live_frame**)
   - **URL**: *https://INSTANCE/live_feed.do?sysparm_doctype=true* where *INSTANCE* is your instance URL (example, *<instance name>.service-now.com*)
   - **Sizing**: select **Fixed Size** and enter height and width pixel dimensions according to the page on which you plan to display the feed (example, width of *1024* and height of *768*)
4. Click **Submit**.
5. Add the block to a page.



iFrame live feed definition

Enhancements

Eureka

- Domain separation is enabled for the Live Feed application. This enhancement adds a **sys_domain** field to Live Feed tables. For more information, see Domain Assignment.

Calgary

The following enhancements were added as of the Calgary release:

- Users can attach any type of supported file to a feed. With previous releases, users could attach images only.
- All live feed properties are now available to administrators in the **Social IT > Feed Administration > Properties module**.
- The **Live Feed** module now appears in the Self-Service application in addition to the Social IT application on the application navigator.

Using Groups in Live Feed

Overview

Groups allow users to create focused discussions in live feed. Users can view or join groups based on the access level of the group:

- **Public:** any user can see the name on the list of all groups, see the feed, and join. Messages appear on the company feed with a link to the group feed.
- **Private:** any user can see the name on the list of all groups, but only invited members can see the feed and join. Messages do not appear on the company feed.
- **Unlisted:** only invited users can see the name on the list of all groups, see the feed, and join. Messages do not appear on the company feed. Private groups may be listed or unlisted.

The group access level is configured by the user who creates the group. The group creator is also the group administrator. Any user can create a group; however, ServiceNow administrators can restrict the ability to create groups.

Note: *The live feed functionality changed significantly with live feed v2 (available with the Fuji release). This page describes live feed v1. For information about live feed v2, see Live Feed.*

Joining Groups (Group Members)

When you join a group, you become a group member. Group members can:

- View messages posted to the group (group feed). Non-members can also view the feed for a public group.
- View a list of group members.
- Invite another user to join the group.
- Subscribe to group email notifications.
- Leave the group.

To join a group:

1. Navigate to **Social IT > Live Feed**.
2. Click **Groups** ( ) and select **View all groups**.
   - If you received an invitation, click **Accept**. You can **Decline** the invitation if you do not want to join the group.
   - If the group is public, click **Join**.
   - If the group is private, click **Request**. The group administrator receives an email notification and must accept your request before you can join the group.

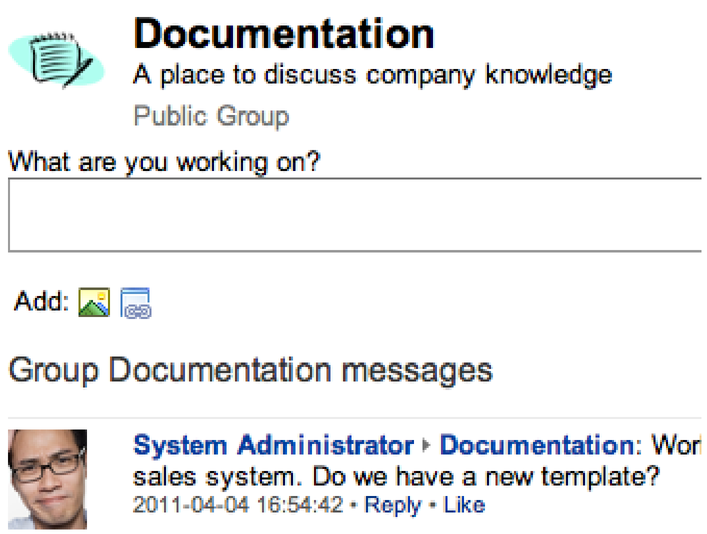
Automatic Group Participation

Users who view records on tables that have live feed enabled, such as the Incident table, automatically join the document group (Calgary Release). If the user creates a record on a table with live feed enabled, the user becomes the document group administrator. Document groups like these are not public. Any user who has access to the record also has access to the document group associated with that record.

Viewing Group Feeds

To view a group feed:

1. Navigate to **Social IT > Live Feed**.
2. Click **Groups** ( ). A list of groups you belong to and an option to view all groups appear.
    - To view a group you belong to, select the group name.
    - To view a public group you do not belong to, select **View all groups** and then click the group name. You must be a member to view the feed for a private group.

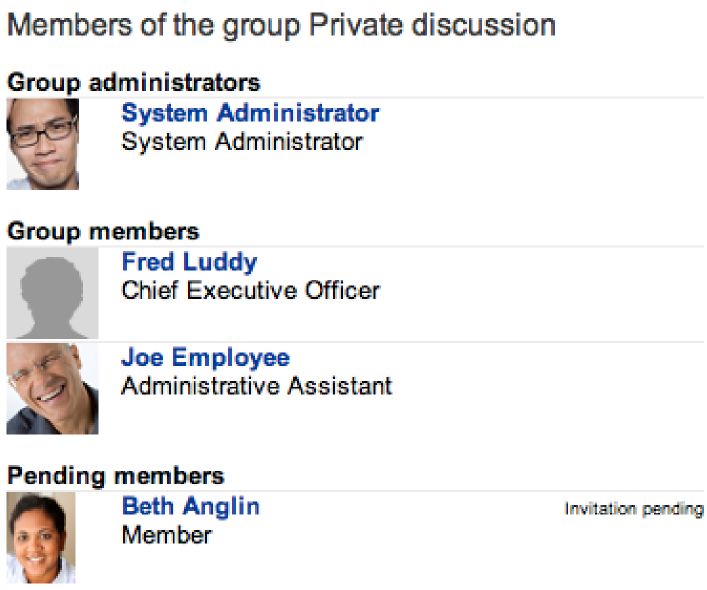


Group feed

Viewing Group Members

To view a list of group members:

1. Navigate to **Social IT > Live Feed**.
2. Click **Groups** ( ) and select the group name.
3. Click **Members** ( ). The list of group members appears, organized by group administrators, group members, and pending members.



Group members list

Inviting New Members

Any group member can invite another user to join a group.

1. Navigate to **Social IT > Live Feed**.
2. Click **Groups** ( ) and select the group name.
3. Click **Invite Member** ( ).
4. Begin entering a user name and select a user from the list, or click the reference lookup icon ( ) and select a user from the table.
5. Click **OK**. The invited user receives an email notification.

Subscribing to Email Notifications

To receive an email notification whenever a message is posted:

1. Navigate to **Social IT > Live Feed**.
2. Click **Groups** ( ) and select the group name.
3. Click **Subscribe** ( ).

You can **Unsubscribe** ( ) from a group to stop receiving email notifications.

Leaving Groups

To leave a group (give up membership):

1. Navigate to **Social IT > Live Feed**.
2. Click **Groups** ( ) and select the group name.
3. Click **Leave**.

Note: *If you leave a document group associated with a record, such as an incident, you automatically rejoin that group if you modify the associated record again (Calgary release).*

Creating Groups (Group Administrators)

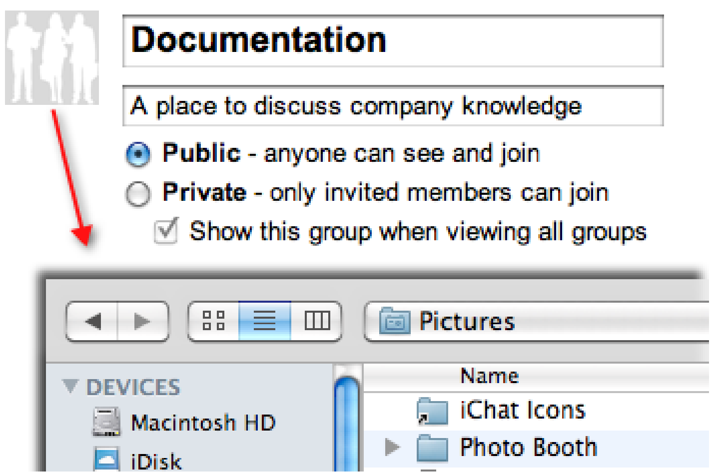
When you create a group, you become the group administrator. Group administrators can:

- Modify group properties
- Accept or reject membership requests
- Delete the group
- Perform all group member functions

To manually create a new group:

1. Navigate to **Social IT > Live Feed**.
2. Click **Create Group** ( ).
3. Enter a **Name** to appear at the top of the group feed and in messages posted to the group (links to the group feed).

4. Enter a **Description** to appear at the top of the group feed and under the group name in the list of all groups.
5. Configure the access level for the group:
   - For a public group, select **Public**.
   - For a private group, select **Private** and select the **Show this group when viewing all groups** check box.
   - For an unlisted group, select **Private** and clear the **Show this group when viewing all groups** check box.
6. [Optional] Select a picture for the group.
   1. Click the picture beside the group name.
   2. Browse to the desired picture file and click **Open**.
7. Click **Create**. The group is created with you as the group administrator.



Create a new group

Automatic Group Creation

If you create records on tables that have live feed enabled, such as the Incident table, a live feed group is automatically created at the same time (Calgary release).

Modifying Groups

To modify an existing group (group administrator only):

1. Navigate to **Social IT > Live Feed**.
2. Click **Groups** ( ) and select the group name.
3. Click **Modify Group** ( ).
4. Modify the group properties or picture.
5. Click **Update**.

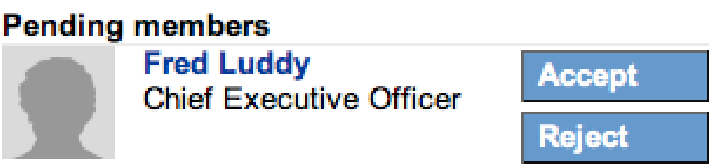
Managing Membership Requests

When a user requests membership for a private group, the group administrator receives an email.

To manage membership requests (group administrator only):

1. Navigate to **Social IT > Live Feed**.
2. Click **Groups** ( ) and select the group name.
3. Click **Members** ( ).
4. In the **Pending Members** section, click **Accept** or **Reject** for each membership request.

   The user receives an email notification to confirm whether the request was accepted or rejected.



Pending membership request

Deleting Groups

To delete a group (group administrator only):

1. Navigate to **Social IT > Live Feed**.
2. Click **Groups** ( ) and select the group name.
3. Click **Delete Group** ( ).

4. Click **Yes** to confirm deletion. The group is deleted from live feed. Messages posted to a private group are deleted; messages posted to a public group remain on the company feed without a group link.

If you delete a group that is associated with an active record, such as an incident that has not yet been closed, the group becomes active again when any user modifies that record or follows the record on live feed. The first user who modifies the record becomes the group administrator. The previous messages that existed as part of the document feed are not deleted (Calgary release).

Administration

Setting Up Live Feed Table Notifications

Overview

Live feed table notifications generate automatic live feed messages. These messages can be generated when specific actions occur in other tables if live feed v2 is active. When a record is inserted or updated on a specific task table and the notification conditions are met, a message is generated and posted to the specified group or the company feed.

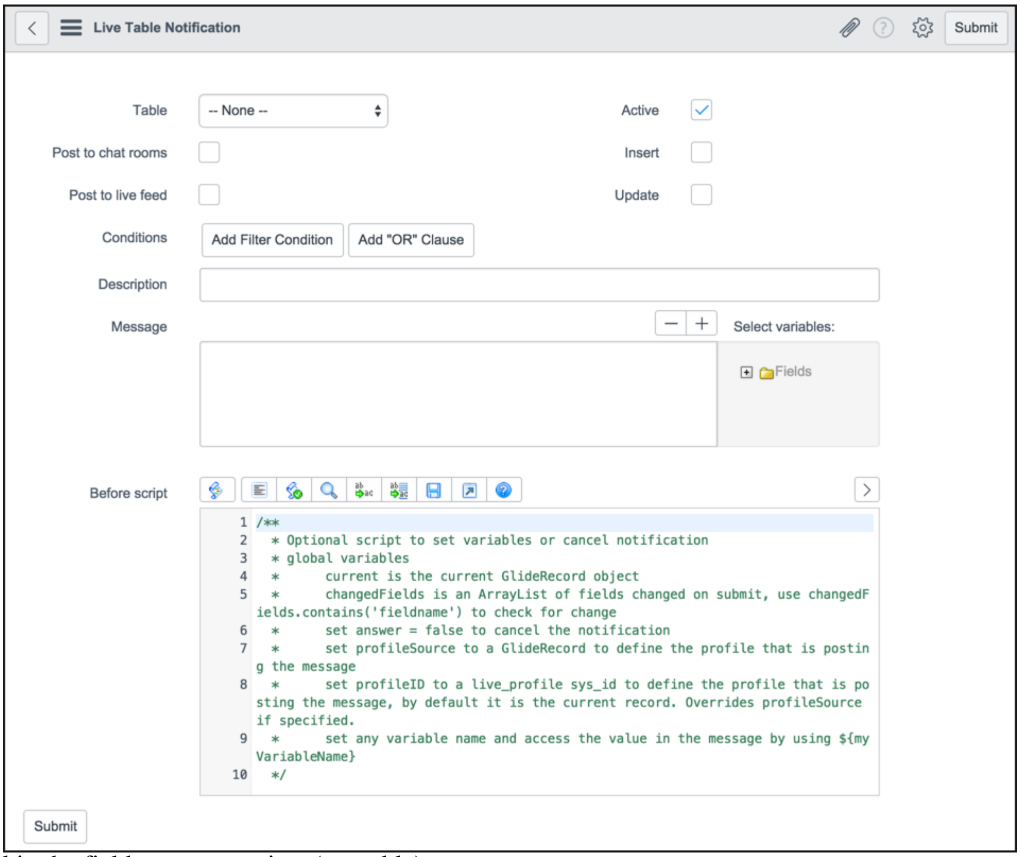
Administrators and users with the live_feed_admin or chat_admin roles can set up table notifications for any table in an instance. To view the table notifications that are available by default, see installed components.

Note: *Many of these features are part of live feed v2, which is available starting with the Fuji release. If you are using live feed v1, or have not activated live feed v2, see Legacy Live Feed.*

Setting Up Table Notifications for Task Tables

To set up a table notification:

1. Navigate to **Social IT > Feed Administration > Table Notifications**.
2. Click **New** or open a notification.



3. Fill in the fields, as appropriate (see table).

4. Click **Submit** or **Update**.

Field	Description
Table	Select the table for which notifications are generated. **Note:** *The list shows only tables and database views that are in the same scope as the table notification (starting with the Fuji release).*
Post to chat rooms	Select the check box to generate the notification in chat rooms that are associated with the record (task records only, requires that the Chat plugin is active).
Post to live feed	Select the check box to generate the notification on live feed. If **Record Feeds** are not set up and the **Feed group** field is blank, the notification is posted to the company feed.
Record Feeds	Enter the record feeds to which notifications are posted. Clear the field to post notifications on the company feed. This field is available only if **Post to live feed** is selected.
Application	Select the application that contains this record.
Active	Select the check box to enable the notification.
Insert	Select the check box to generate the notification when a record is inserted into the database.
Update	Select the check box to generate the notification when a record is updated.
Feed	Select the live feed team to which you want to post messages. This field is available only if **Post to live feed** is selected.
Conditions	[Optional] Create the condition that must be met to generate the notification. If you add a condition statement, the system evaluates the condition first and parses the **Before script** field only if the condition is met. You may choose to leave this field blank and include conditions in the script.
Description	Enter a description of the table notification.

Message	Construct the automatic message you want to generate. You can include field values as variables and a link to the record. To add a variable, position the cursor in the text and then click the field name in the **Select variables** field. To add a link to the record, enter **${URI}** in the message text. The following is an example of message syntax:
	`High priority incident opened: ${URI}`
	`Short description: ${short_description}`
	Note: For security reasons, any HTML code is automatically stripped out of a message before it is posted.

Before script	[Optional] Create a script that runs before a notification is generated. The script runs only if the condition evaluates to **true** or is empty. The following variables are available in the script: • **current**: the GlideRecord object that triggered the notification. • **changedFields**: an ArrayList of fields changed on submit. Use changedFields.contains('fieldname') to check for change. • **answer**: set answer to false to cancel the notification. • **profileSource**: set to a valid GlideRecord object to define the profile that is posting the message. • **profileID**: set to a live_profile sys_id to define the profile that is posting the message; by default it is the current record. Overrides profileSource if both are specified. Any custom variable created in the script is also exposed to the message in the form of `${myVariableName}`.

Setting Up Table Notifications for Additional (Non-Task) Tables

Table notifications are supported for all task tables by default. If you have the live_feed_admin or chat_admin role, you can set up a table notification for any other table.

To set up a table notification for an additional, non-task, table:

1. Navigate to **System Definition > Business Rules**.
2. Open the **live feed events** business rule.
3. Select and copy the text in the **Script** field.
4. Navigate to **System Definition > Business Rules**.
5. Click **New**.
6. Enter the following values at the top of the form:
   - **Name**: Enter a name, such as *live feed events for my table*.
   - **Table**: Select the table for which you are setting up a notification.
   - **Active** and **Advanced:** Select these check boxes.
7. In the **When to run** section, enter the following values:
   - **When**: Enter *before*.
   - **Insert**, **Update**, and **Delete**: Select the check boxes.
8. In the **Advanced** section, paste the script from the **live feed events** business rule (step 3) into the script box.
9. Click **Submit**.

   The **live feed events** business rule runs on the non-task table. It fires the **live_feed.update** event, which is associated with the **Live Feed Update** script action. The script action sets up variables and processes the table notifications.
10. Create a live feed table notification record for the new table as you would for a task table.

Setting Up Table Notifications for Related Feeds

In addition to being able to set up table notifications for a specific feeds, administrators can also set up table notifications to be sent out to related feeds. For example, a table notification can be set up so that whenever the status of a story in the SDLC application changes, a message is sent to the related sprint, release, or epic. The messages are posted only if the related feed already exists; it will not create a new feed.

For additional details, see Example 4. Table Notification to Related Feeds.

Examples

Example 1. Workaround Posted

This example demonstrates a table notification that generates an automatic message on live feed whenever a workaround is added to an open problem.

- **Table**: *Problem [problem]*
- **Active**: Select the check box.
- **Update**: Select the check box.
- **Post to live feed**: Select the check box.
- **Conditions**: *State is Open*
- **Description**: *Workaround Posted*
- **Message**:

```
${sys_updated_by} posted a workaround for ${URI}.
Short description: ${short_description}
```

- **Before script**:

```
// only post to live feed when the Workaround field changes
answer = changedFields.contains("work_around");
```

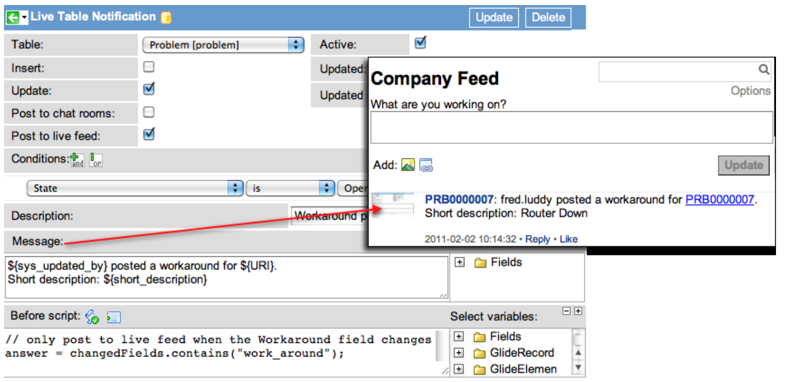


Table notification. Posts a message to live feed when a workaround is posted on an open problem.

Example 2. Problem Resolved (Advanced)

This advanced example demonstrates a table notification that generates an automatic message on live feed whenever a problem is closed. It also adds a message about the assigned user and posts the message from the assignment group profile instead of the problem record.

- **Table**: *Problem [problem]*
- **Active**: Select the check box.
- **Update**: Select the check box.
- **Post to live feed**: Select the check box.
- **Conditions**: *Problem State is Closed/Resolved*
- **Description**: *Problem Resolved*
- **Message**:

```
Problem ${number} - ${short_description} has been resolved. ${fixedByMsg}
```

- **Before script**:

```
//cancel if we didn't just change the problem state
if (!changedFields.contains("problem_state"))
answer = false;

//if we have an assigned_to value add a comment about who it was
//create a new variable fixedByMsg that we can access from the message
fixedByMsg = "";
if (!current.assigned_to.nil())
fixedByMsg = " Thank you " + current.assigned_to.getDisplayValue();

//make the message appear to come from the assignment group if we have
one
if (!current.assignment_group.nil())
profileSource = current.assignment_group.getRefRecord(); //need
GlideRecord object
```

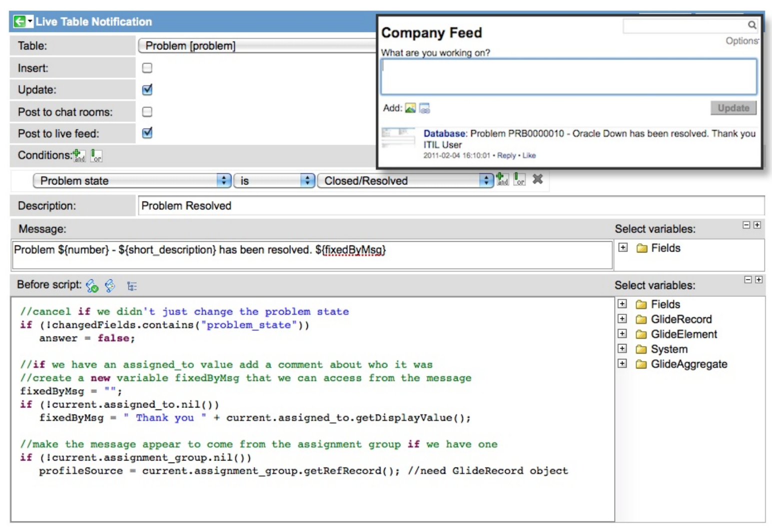


Table notification. Posts a message to live feed when a problem is resolved.

Example 3. Opportunity Won (Non-Task)

This example demonstrates a table notification on a non-task table. It generates an automatic message on live feed whenever a sales opportunity is won (requires the Sales Force Automation plugin).

Create a business rule on the Opportunity [sales_opportunity] table that matches the **live feed events** business rule on the task table.

Create a table notification with the following values:

- **Table**: *Opportunity [sales_opportunity]*
- **Active**: Select the check box.
- **Update**: Select the check box.
- **Post to live feed**: Select the check box.
- **Conditions**: *State is Closed Won*
- **Description**: *Opportunity won*
- **Message**:

```
${owner} closed a sale with ${account}!
```

- **Before script**:

```
//make the message appear to come from the assigned salesperson if
there is one
if (!current.owner.nil())
profileSource = current.owner.getRefRecord(); //need GlideRecord object
```

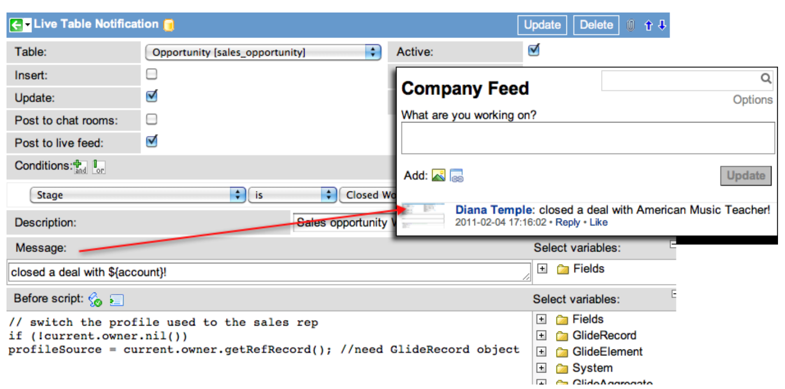


Table notification. Posts a message to live feed when a sales opportunity is won.

Example 4. Table Notification to Related Feeds

This example demonstrates table notifications to be sent out to related feeds. For this example, a table notification is set up so that whenever the status of a story changes to **Complete**, a message is sent to the related sprint, release, or epic. Messages are posted only if the related feed already exists; this notification does not create a new feed.

Create a table notification with the following values:

- **Table**: *Story [rm_story]*
- **Active**: Select the check box.
- **Update**: Select the check box.
- **Post to live feed**: Select the check box.
- **Record Feeds**: move **Sprint**, **Release**, and **Epic** to the **Selected** column.
- **Conditions**: *State changes to Complete*
- **Description**: *Story is done; message to Epic, Release and Sprint*
- **Message**:

```
${URI} status changed to ${state}
```

Article Sources and Contributors

Social IT *Source*: http://wiki.servicenow.com/index.php?oldid=250902 *Contributors*: Cheryl.dolan, Emily.partridge, John.ramos, Joseph.messerschmidt, Rachel.sienko

Live Feed *Source*: http://wiki.servicenow.com/index.php?oldid=250693 *Contributors*: Emily.partridge, Fuji.publishing.user, Joe.zucker, John.ramos, John.roberts, Joseph.messerschmidt, Julie.phaviseth, Liz.malone, Peter.smith, Phillip.salzman, Rachel.sienko

Using Live Feed *Source*: http://wiki.servicenow.com/index.php?oldid=243362 *Contributors*: Fuji.publishing.user, Peter.smith

Using Groups in Live Feed *Source*: http://wiki.servicenow.com/index.php?oldid=246196 *Contributors*: Fuji.publishing.user, Peter.smith

Setting Up Live Feed Table Notifications *Source*: http://wiki.servicenow.com/index.php?oldid=241616 *Contributors*: Emily.partridge, Fuji.publishing.user, John.roberts, Joseph.messerschmidt, Peter.smith, Rachel.sienko

Image Sources, Licenses and Contributors

Image:Warning.gif *Source*: http://wiki.servicenow.com/index.php?title=File:Warning.gif *License*: unknown *Contributors*: CapaJC

Image:SocialITapp.png *Source*: http://wiki.servicenow.com/index.php?title=File:SocialITapp.png *License*: unknown *Contributors*: Rachel.sienko

Image:My_Live_Feeds.png *Source*: http://wiki.servicenow.com/index.php?title=File:My_Live_Feeds.png *License*: unknown *Contributors*: Fuji.publishing.user

Image:live_feed_menu.png *Source*: http://wiki.servicenow.com/index.php?title=File:Live_feed_menu.png *License*: unknown *Contributors*: Fuji.publishing.user

Image:my_feed.png *Source*: http://wiki.servicenow.com/index.php?title=File:My_feed.png *License*: unknown *Contributors*: Fuji.publishing.user, Phillip.salzman

Image:SocialITLiveFeedApp.png *Source*: http://wiki.servicenow.com/index.php?title=File:SocialITLiveFeedApp.png *License*: unknown *Contributors*: Fuji.publishing.user, Phillip.salzman, Rachel.sienko

Image:LiveiFrame.png *Source*: http://wiki.servicenow.com/index.php?title=File:LiveiFrame.png *License*: unknown *Contributors*: Fuji.publishing.user, Rachel.sienko

Image:GroupsList.png *Source*: http://wiki.servicenow.com/index.php?title=File:GroupsList.png *License*: unknown *Contributors*: Rachel.sienko

Image:GroupFeed.png *Source*: http://wiki.servicenow.com/index.php?title=File:GroupFeed.png *License*: unknown *Contributors*: Rachel.sienko

Image:contacts.png *Source*: http://wiki.servicenow.com/index.php?title=File:Contacts.png *License*: unknown *Contributors*: Rachel.sienko

Image:GroupMembers.png *Source*: http://wiki.servicenow.com/index.php?title=File:GroupMembers.png *License*: unknown *Contributors*: Rachel.sienko

Image:Plus.png *Source*: http://wiki.servicenow.com/index.php?title=File:Plus.png *License*: unknown *Contributors*: CapaJC, Rachel.sienko

Image:SearchIcon.png *Source*: http://wiki.servicenow.com/index.php?title=File:SearchIcon.png *License*: unknown *Contributors*: CapaJC, Rachel.sienko

Image:Subscribe.png *Source*: http://wiki.servicenow.com/index.php?title=File:Subscribe.png *License*: unknown *Contributors*: Rachel.sienko

Image:Unsubscribe.png *Source*: http://wiki.servicenow.com/index.php?title=File:Unsubscribe.png *License*: unknown *Contributors*: Fuji.publishing.user, Rachel.sienko

Image:CreateGroup.png *Source*: http://wiki.servicenow.com/index.php?title=File:CreateGroup.png *License*: unknown *Contributors*: Rachel.sienko

Image:GroupProperties.png *Source*: http://wiki.servicenow.com/index.php?title=File:GroupProperties.png *License*: unknown *Contributors*: Rachel.sienko

Image:ModifyGroup.png *Source*: http://wiki.servicenow.com/index.php?title=File:ModifyGroup.png *License*: unknown *Contributors*: Rachel.sienko

Image:PendingMembership.png *Source*: http://wiki.servicenow.com/index.php?title=File:PendingMembership.png *License*: unknown *Contributors*: Rachel.sienko

Image:Delete_row.png *Source*: http://wiki.servicenow.com/index.php?title=File:Delete_row.png *License*: unknown *Contributors*: Rachel.sienko

image:table_notification.png *Source*: http://wiki.servicenow.com/index.php?title=File:Table_notification.png *License*: unknown *Contributors*: Fuji.publishing.user

Image:TableNotification.png *Source*: http://wiki.servicenow.com/index.php?title=File:TableNotification.png *License*: unknown *Contributors*: Rachel.sienko

Image:live_notification_adv_example.jpg *Source*: http://wiki.servicenow.com/index.php?title=File:Live_notification_adv_example.jpg *License*: unknown *Contributors*: John.roberts

Image:TableNotificationSales.png *Source*: http://wiki.servicenow.com/index.php?title=File:TableNotificationSales.png *License*: unknown *Contributors*: Rachel.sienko