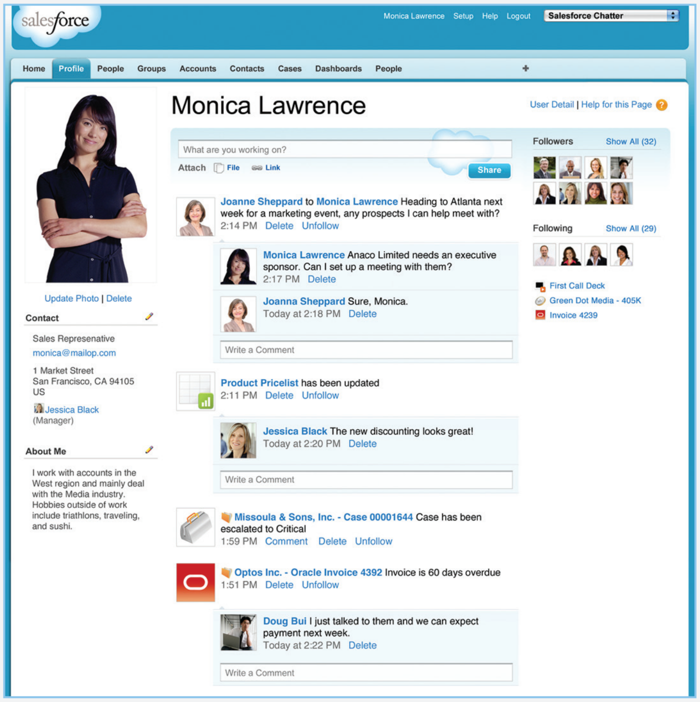
Chatter Cheat Sheet

Overview

This cheat sheet describes the data model that supports the Salesforce Chatter collaboration platform. Chatter provides profiles, status updates, posts, comments, feeds, and groups - and each of these items can be found in the data model, exposed on Force.com. Using this data model, you can extend your own applications with Chatter support, or build new ones with the Chatter platform.



Monica Lawrence

Joanne Sheppard to Monica Lawrence Heading to Atlanta next week for a marketing event, any prospects I can help meet with?
2:14 PM Delete Unfollow

Monica Lawrence Anaco Limited needs an executive sponsor. Can I set up a meeting with them?
2:17 PM Delete

Joanna Sheppard Sure, Monica.
Today at 2:18 PM Delete

Product Pricelist has been updated
2:11 PM Delete Unfollow

Jessica Black The new discounting looks great!
Today at 2:20 PM Delete

Missoula & Sons, Inc. - Case 00001644 Case has been escalated to Critical
1:59 PM Comment Delete Unfollow

Optos Inc. - Oracle Invoice 4392 Invoice is 60 days overdue
1:51 PM Delete Unfollow

Doug Bui I just talked to them and we can expect payment next week.
Today at 2:22 PM Delete

Sales Represenative
monica@mailop.com

1 Market Street
San Francisco, CA 94105
US

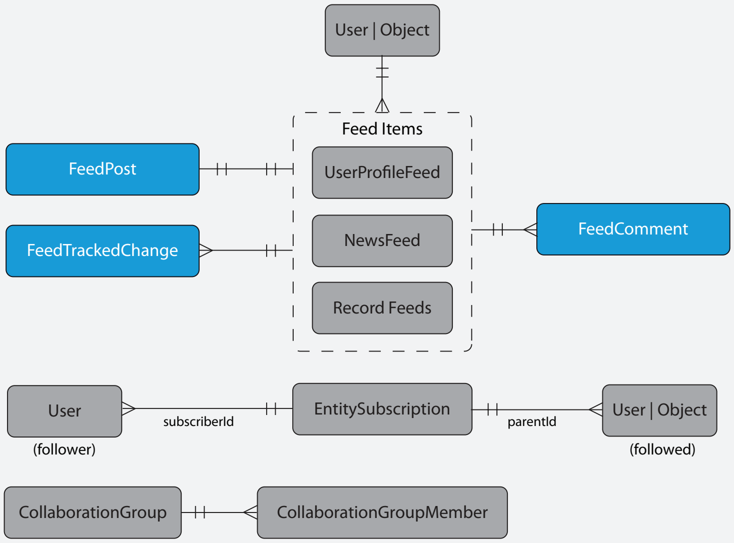
Jessica Black
(Manager)

I work with accounts in the West region and mainly deal with the Media industry. Hobbies outside of work include triathlons, traveling, and sushi.

First Call Deck
Green Dot Media - 405K
Invoice 4239

Chatter Data Model

The following diagram depicts the most important objects found in the Chatter data model.



Three primary objects provide a route to most Chatter data:

- The UserProfileFeed aggregates posts and tracked changes a user has made
- The NewsFeed aggregates posts on users and records that a user is following
- Record feeds, one for each object that is enabled for feed tracking, aggregate posts that indicate changes to record data

Collectively, the items in these feeds are called feed items.

Not all data about a post is stored in a feed item. Comments are stored in FeedComment records, the details about a tracked change are stored in FeedTrackedChange records, and the data in a post (such as the text of the post itself) is stored in a FeedPost record. None of these three objects can be queried directly. Instead, query them through a UserProfileFeed, NewsFeed or record feed.

Feed Items

Though not an object, feed items is a term for the records found in a NewsFeed, UserProfileFeed or record feeds. Each feed item is associated with either a set of tracked changes to a record, or a FeedPost.

The Type field indicates whether the feed item represents a UserStatus, TextPost, LinkPost, ContentPost or TrackedChange. If it's of type TrackedChange, additional data is in the FeedTrackedChange object. If it's of any other type, FeedPost holds the additional data.

Each feed item represents either a set of changes on a specific record or a post to a particular user or record, indicated by its ParentId field. For instance, when you update your status, the ParentId of the resulting feed item, holds your UserId. Note that some queries and statements, for example adding a comment, require the ID of a feed item. Fields include:

- Type - one of UserStatus, TextPost, LinkPost, ContentPost or TrackedChange
- CreatedDate - when the feed item was created
- CreatedBy - the user who made the change or post
- ParentId - the ID of the user or record about which the feed item was created

FeedPost

Stores most information about a post. Depending on type, different fields will be populated. All FeedPost records contain a FeedItemId field pointing back to the feed item. Query this object only through a UserProfileFeed, NewsFeed or record feed. Fields include:

- FeedItemId - ID of the item containing the FeedPost
- Body - required content of a TextPost or UserStatus, optional for other types
- LinkURL - URL of a LinkPost
- Title - title of a link in a LinkPost
- ContentType - The MIME type of the attachment for ContentPosts (read only)
- ContentSize - the size of the attachment for ContentPosts (read only)
- ContentFileName - the file name of the attachment for ContentPosts
- ContentDescription - description of the attachment for ContentPosts
- ContentData - Base64-encoded attachment for ContentPosts

While you cannot read FeedPost records directly, you can insert them. ParentId can point to a user or a record that is tracked.

Create a new Text Post

```
FeedPost fpost = new FeedPost();
fpost.Type = 'TextPost';
fpost.ParentId = <Id of User or Record>;
fpost.Body = 'The mice will see you now';
insert fpost;
```

Other Examples

See the following queries for additional examples:

- Home Tab Query
- Record Detail Page Query
- Profile Tab Query

Record Feeds

An object with feed tracking enabled, has changes to tracked fields aggregated by feed items in its record feed. Each such object has its own record feed. Record feeds for standard objects are named after the object, so Account has an AccountFeed object representing its record feed, while custom objects like Foo__c have a record feed named Foo_Feed__c. The actual details of the field change are stored in the FeedTrackedChange object.

Fields: FeedPostId (ID of associated FeedPost), ParentId (ID of record being tracked), Type

OpportunityFeed Feed Items on a Record

```
ID OpportunityId = <Id of an opportunity>;
List<OpportunityFeed> oRFeed = [
  SELECT Id, Type, CreatedBy.Name, Parent.Name, FeedPost.Body
  FROM OpportunityFeed WHERE ParentId = :OpportunityId
  ORDER BY CreatedDate DESC, Id DESC LIMIT 20
];
```

Record Detail Page Query

```
ID accountId = <Id of an Account>;
List<AccountFeed> aRFeed = [
  SELECT Id, Type, CreatedDate, CreatedById, CreatedBy.Name,
    ParentId, Parent.Name, FeedPostId, FeedPost.Body,
    FeedPost.Title, FeedPost.LinkUrl,
    (SELECT Id, FieldName, OldValue, NewValue
      FROM FeedTrackedChanges),
    (SELECT Id, CreatedDate, CreatedById,
        CreatedBy.Name, CommentBody
      FROM FeedComments ORDER BY CreatedDate DESC)
  FROM AccountFeed WHERE ParentId = :accountId
  ORDER BY CreatedDate DESC, Id DESC LIMIT 10
];
```

This query is used to display the items on an account record's detail page. This query works for any record feed - simply replace AccountFeed with a different record feed. See Profile Tab Query for a description of the return values.

NewsFeed

This object aggregates feed items that represent a user's posts, as well as posts on users and records that the user is following. This feed displays on the Home tab in the Salesforce Chatter application. Think of it as feed items whose ParentId is the user, or whose ParentId is in the set of users or records the user follows (represented by EntitySubscription records). You can only query your own NewsFeed.

Fields: FeedPostId (ID of associated FeedPost), ParentId (ID of the user or record being tracked), Type

Notes:

- If a follower, Jack, posted on a record you follow, Acme, then ParentId points to Acme
- Use the CreatedBy system field to determine who made the post

NewsFeed Feed Items on Users and Records I Follow

```
List <NewsFeed> aNewsFeed = [
  SELECT Id, Type, CreatedBy.Name, Parent.Name, FeedPost.Body
    FROM NewsFeed
    ORDER BY CreatedDate DESC, Id DESC LIMIT 20
];
```

This lists an abbreviated form of the Chatter feed on your Home tab. CreatedBy.Name indicates who created the item. Parent.Name is your name (if someone posted on your User record) or a record or user's name (if someone posted on a record or user you follow), while FeedPost.Body is non-null if the item isn't of type TrackedChange.

Home Tab Query

```
List <NewsFeed> aNewsFeed = [
  SELECT Id, Type, CreatedDate, CreatedById, CreatedBy.Name,
    ParentId, Parent.Name, FeedPostId, FeedPost.Body,
    FeedPost.Title, FeedPost.LinkUrl,
    (SELECT Id, FieldName, OldValue, NewValue
      FROM FeedTrackedChanges),
    (SELECT Id, CreatedDate, CreatedById,
        CreatedBy.Name, CommentBody
      FROM FeedComments ORDER BY CreatedDate DESC)
  FROM NewsFeed ORDER BY CreatedDate DESC, Id DESC LIMIT 10
];
```

This query is used to display the items on the Home tab. See Profile Tab Query for a description of the return values.

UserProfileFeed

This object aggregates feed items that represent all posts and all tracked changes that a user has made. This feed displays in the Profile tab of the Salesforce Chatter application. Think of it as feed items that have a CreatedBy or ParentId mapping to the user.

Fields: FeedPostId (ID of the associated FeedPost), ParentId (ID of the user or record), Type

Notes:

- When querying this feed, you must include a WITH clause and specify the UserId of the user whose profile you want to query
- FeedPostId is null if the instance of UserProfileFeed represents a record update

UserProfileFeed Feed Items indicating Posts a User has Made

```
ID userID = <id of User>;
List <UserProfileFeed> uposts  = [
  SELECT Id, FeedPost.Body, Type, Parent.Name
    FROM UserProfileFeed WHERE Type = 'TextPost'
    WITH UserId=:userId
    ORDER BY CreatedDate DESC, Id DESC LIMIT 10
];
```

This lists the 10 most recent feed items that are associated with text posts, and the associated post contents. Note the WITH clause, used to specify which user's profile to query. If you omit the WHERE clause, then all types of feed items are returned, including tracked changes. For tracked changes, FeedPost is null. ParentId points to the record that was updated.

Profile Tab Query

```
ID userID = <id of User>;
List <UserProfileFeed>  = [
  SELECT Id, Type, CreatedDate, CreatedById, CreatedBy.Name,
      ParentId, Parent.Name, FeedPostId, FeedPost.Body,
      FeedPost.Title, FeedPost.LinkUrl,
      (SELECT Id, FieldName, OldValue, NewValue
        FROM FeedTrackedChanges),
      (SELECT Id, CreatedDate, CreatedById,
                CreatedBy.Name, CommentBody
        FROM FeedComments ORDER BY CreatedDate DESC)
  FROM UserProfileFeed WITH UserId = :userId
  ORDER BY CreatedDate DESC, Id DESC LIMIT 10
];
```

This query is used to display the items on the Profile tab. The nested selects return a set of records for each UserProfileFeed record. If there are comments on the feed item, FeedComments represents those comments, otherwise it is null. If the feed item is of type TrackedChange, there are one or more FeedTrackedChange records associated with the feed item, representing all of the tracked field changes made during a single update to a record. The user running the query may not have field-level security access to all the fields that were changed, and so may not have access to the corresponding FeedTrackedChanges records either. As a result, the set of FeedTrackedChanges might be null for that user.

FeedTrackedChange

This object stores individual field changes on records that belong to objects that have feed tracking enabled. Query this object only through a UserProfileFeed, NewsFeed or record feed.

Fields: Id, FieldName (name of the field that was changed), OldValue, NewValue, FeedItemId (ID of the feed item that represents the record feed item on which the field changed)

Tracked Changes to a Record

```
ID OpportunityId = <Id of an opportunity>;
List<OpportunityFeed> oEFeedT =  [
  SELECT Id, Type, CreatedBy.Name, Parent.Name,
    (SELECT OldValue, NewValue FROM FeedTrackedChanges)
  FROM OpportunityFeed
  WHERE ParentId = :OpportunityId AND Type = 'TrackedChange'
  ORDER BY CreatedDate DESC, Id DESC LIMIT 20
];
```

Because ParentId points to an opportunity record, this query returns all OpportunityFeed (a record feed) items about that opportunity.

FeedComment

This object stores comments. A FeedComment is a child object of an associated UserProfileFeed, NewsFeed, or record feed item. Each FeedComment is associated with a feed item. Query this object only through a UserProfileFeed, NewsFeed or record feed.

Fields: CommentBody (the string representing the comment's text), FeedItemId (ID of the feed item on which the comment was made), ParentId (ID of the record associated with the comment)

Note:

- ParentId either points to a record on which a comment was left, or a User on whose post a comment was left - functionally equivalent to the associated feed item's parent.

Create a Comment

```
FeedComment fcomment = new FeedComment();
fcomment.FeedItemId = <feedItemId>;
fcomment.CommentBody = 'This is a profound comment.';
insert fcomment;
```

You need the ID of a feed item for this to work. All the queries on the UserProfileFeed, NewsFeed and record feeds return feed item records.

Other Examples

See the following queries for additional examples:

- Home Tab Query
- Record Detail Page Query
- Profile Tab Query

EntitySubscription

This object represents a subscription of a user. Use it to determine which users or records a user is following, or who is following a user or record.

Fields: ParentId (ID of the user or record being followed), SubscriberId (ID of the user following the record)

Followers of a User or Record

```
ID uid = <Id of User or Record>;
List<EntitySubscription> followers = [
  SELECT Id, SubscriberId, Subscriber.Name
  FROM EntitySubscription WHERE ParentId = :uid
];
```

List the Users and Records I Follow

```
ID uid = <Id of User>;
EntitySubscription[] followingES = [
  SELECT Id, ParentId, SubscriberId, Parent.Name
  FROM EntitySubscription WHERE SubscriberId = :uid
];
```

To only show users, add Parent.Type='User' to the WHERE clause.

List Subscriptions to Records

```
ID uid = <Id of User>;
List<EntitySubscription> les = [
  SELECT Id, Parent.Name FROM EntitySubscription
  WHERE SubscriberId = :uid AND ParentId IN
      (SELECT Id FROM Account WHERE NumberOfEmployees >= 10)
];
```

This lists accounts a user follows that have more than 10 employees.

CollaborationGroup

This object represents Chatter groups.

Fields: Name, Description, OwnerId, CollaborationType (Public or Private)

The CollaborationGroup object has an associated record feed called CollaborationGroupFeed, which aggregates the feed items associated with a group. Use CollaborationGroupFeed as you would any other record feed.

List Groups

```
List<CollaborationGroup> oG = [SELECT Name, Description,
CollaborationType, OwnerId from CollaborationGroup];
```

This query returns a list of all Chatter groups. The CollaborationType field will either be Private or Public.

List a Group's Feed

```
ID cgid = <Id of a group>;
List<CollaborationGroupFeed> ic = [
  SELECT Id, Type, ParentId, Parent.Name, FeedPost.Id,
      FeedPost.Type , FeedPost.Body, FeedPost.Title,
      (SELECT Id,  FieldName, OldValue, NewValue
        FROM FeedTrackedChanges ORDER BY Id DESC),
      (SELECT Id, CommentBody, CreatedDate, CreatedById,
        CreatedBy.FirstName, CreatedBy.LastName
        FROM FeedComments
        ORDER BY CreatedDate DESC, Id DESC LIMIT 4)
  FROM CollaborationGroupFeed WHERE ParentId = :cgid
  ORDER BY CreatedDate DESC, Id DESC LIMIT 20
];
```

Changes to a group's name, description or similar fields appear as tracked changes on the group's feed. This query ensures the tracked change values are retrieved, as well as comments on posts.

CollaborationGroupMember

This object associates Chatter group members with Chatter groups.

Fields: CollaborationGroupId (Id of the CollaborationGroup), MemberId (ID of the user who is a member)

List Members of a Group

```
ID cg = <Id of a Chatter group>;
List<CollaborationGroupMember> oM =  [
  SELECT MemberId, CollaborationGroup.Name
  FROM CollaborationGroupMember WHERE CollaborationGroupId = :cg
];
```

This query returns the IDs of members for a particular Chatter group.

Add a User to a Group

```
ID uid = <Id of a user>;
ID cgid = <Id of a group>;
CollaborationGroupMember cgm = new CollaborationGroupMember(
  memberid = uid, collaborationgroupid = cgid);
insert cgm;
```

CurrentStatus

Update a user’s status by updating the CurrentStatus field of the User record. Updating the field will also ensure that UserProfileFeed is updated.

Update a User's Status

```
ID uid = <Id of User>;
User user = [SELECT Id, CurrentStatus FROM User WHERE Id = :uid];
user.CurrentStatus = 'new and exciting status message';
update u;
```

This retrieves a User record, sets the CurrentStatus field, and updates the record.

chatter

developerforce™

Go to http://developer.force.com/cheatsheets for additional cheat sheets and resources.

Corporate Headquarters
The Landmark @ One Market
Suite 300
San Francisco, CA, 94105
United States

1-800-NO-SOFTWARE
www.salesforce.com

Latin America
+1-415-536-4606

Japan
+81-3-5785-8201

Asia/Pacific
+65-6302-5700

Europe, Middle East & Africa
+4121-6953700