



Salesforce.com: Spring '12

Database.com Spring '12 Release Notes



Last updated: March 18 2012

© Copyright 2000–2012 salesforce.com, inc. All rights reserved. Salesforce.com is a registered trademark of salesforce.com, inc., as are other names and marks. Other marks appearing herein may be trademarks of their respective owners.

Table of Contents

About the Release Notes.....	2
Release Notes Changes.....	3
Summary of Spring '12 Features and Impact on Database.com Users.....	4
Database.com Console Enhancements.....	4
Security Enhancements.....	5
Sharing Enhancements.....	6
Apex Code Enhancements.....	6
API Enhancements.....	7
Additional Database.com Enhancements.....	7
Database.com Console Enhancements.....	8
Schema Builder Enhancements—Beta.....	8
Platform Development Tools.....	12
Security Enhancements.....	15
Sharing Enhancements.....	16
Apex Code Enhancements.....	19
API Enhancements.....	23
New and Changed Objects.....	23
SOQL Enhancements.....	25
SOAP Web Services API Enhancements.....	26
REST API Enhancements.....	27
Streaming API Enhancements.....	27
Metadata API Enhancements.....	28
Additional Database.com Enhancements.....	30
Help and Training Enhancements.....	32

About the Release Notes

The Release Notes are a comprehensive user guide for the latest release of Database.com. Unlike a traditional release notes document that includes only a simple list of enhancements, the Database.com Release Notes give you everything you need to get up and running with the new features and enhancements in the latest release.

What's Included in the Release Notes

For every new major enhancement, the Release Notes provide:

- A brief, high-level description of the functionality
- Implementation tips to help you get started with setup and administration
- Best practice tips to help you maximize the benefit of the functionality
- Complete end-to-end instructions on how to set up and use the functionality

Beyond the major new features, the Additional Enhancements sections include a list and brief description of every other enhancement or functional change included in the latest release—everything from email enhancements, to new report types, to security and packaging enhancements.

Let the Release Notes be your guide to success with the latest release from salesforce.com!

Your Feedback Matters

We know how important the Release Notes, online help, and documentation are to your company's success with Database.com. To continually improve the content we deliver to you, we want to know what works and what doesn't. Let us know!

- Feedback forms—Every HTML documentation page, both in the online help and in our developer guides at [Developer Force](#), includes a feedback form for you to submit your suggestions, corrections, and feedback about the documentation. Let us know what you think!
- IdeaExchange—We're listening to your ideas too. Spring '12 includes some of your top ideas. Visit [IdeaExchange](#) for a complete list of ideas coming in Spring '12.

Want to be notified whenever we publish new documentation or make significant updates to existing documentation? Follow us on Twitter: [@salesforcedocs](#).

Release Notes Changes

You asked for it, we listened. In Spring '12, we're introducing the Release Notes Change Log. No more wondering what's changed with each new iteration of the release notes. No more sifting through the sand to find the gold nuggets. It's all right here, with the newest updates at the top.

Date	What's New	Description
February 22, 2012	Name Fields for Groups, Queues, and Roles	Updated information about name fields for sharing entities.
February 22, 2012	Name Fields for Sharing Rules	Updated information about name fields for sharing rules.
February 1, 2012	Joined Reports—Generally Available, Cross Filters—Generally Available, and Data Bucketing—Generally Available	Updated information about joined reports, cross filters, and data bucketing to clarify that they are included automatically in Enterprise and Unlimited editions, and to provide information about their release schedules.
January 31, 2012	Automatic Deployment of New Custom Components	Added an entry explaining auto-deployment of new custom objects, report types, and workflow rules in an installed or upgraded package.
January 31, 2012	Streaming API—Generally Available	Added an entry announcing that Streaming API is now generally available.
January 24, 2012	Help and Training Enhancements	Added Help and Training Enhancements section.
January 4, 2012	Release Notes Change Log	New feature to track updates and changes to the release notes.

Summary of Spring '12 Features and Impact on Database.com Users

Spring '12 has features that immediately impact all users after the release. You might want to communicate these changes to your users beforehand so they are prepared. Other features require direct action by an administrator before users can benefit from the new functionality.

See Also:

[Database.com Console Enhancements](#)

[Security Enhancements](#)

[Sharing Enhancements](#)

[Apex Code Enhancements](#)

[API Enhancements](#)

[Additional Database.com Enhancements](#)

Database.com Console Enhancements

Feature	Automatically visible to all users. No setup required.	Automatically visible to all administrators. No setup required.	Not automatically visible. Feature is available but requires some setup.	Contact salesforce.com to enable this feature.
Database.com Console Enhancements	✓			
Schema Builder Enhancements—Beta	✓			

Platform Development Tools—Developer Console

Feature	Automatically visible to all users. No setup required.	Automatically visible to all administrators. No setup required.	Not automatically visible. Feature is available but requires some setup.	Contact salesforce.com to enable this feature.
Raw Debug Log Viewer		✓		
Heap Search		✓		

Feature	Automatically visible to all users. No setup required.	Automatically visible to all administrators. No setup required.	Not automatically visible. Feature is available but requires some setup.	Contact salesforce.com to enable this feature.
General Development Improvements		✓		

See Also:

[Summary of Spring '12 Features and Impact on Database.com Users](#)

Security Enhancements

Feature	Automatically visible to all users. No setup required.	Automatically visible to all administrators. No setup required.	Not automatically visible. Feature is available but requires some setup.	Contact salesforce.com to enable this feature.
Single Sign-On Changes		✓		
Increased Maximum Timeout Value		✓		
New Password Complexity Option		✓		
Requiring a Secure Connection (HTTPS) Is Permanent		✓		
Increased Service Provider Selection Availability	✓			
Encrypted Fields Available in Staging Database				✓
Network Access Column Changes	✓			

See Also:

[Summary of Spring '12 Features and Impact on Database.com Users](#)

Sharing Enhancements

Feature	Automatically visible to all users. No setup required.	Automatically visible to all administrators. No setup required.	Not automatically visible. Feature is available but requires some setup.	Contact salesforce.com to enable this feature.
Defer Sharing Calculations (available within 24 hours after the Spring '12 release)				✓
Name Fields for Groups, Queues, and Roles	✓			
Name Fields for Sharing Rules	✓			
Sharing Rule Components for Change Sets	✓			

See Also:

[Summary of Spring '12 Features and Impact on Database.com Users](#)

Apex Code Enhancements

Feature	Automatically visible to all users. No setup required.	Automatically visible to all administrators. No setup required.	Not automatically visible. Feature is available but requires some setup.	Contact salesforce.com to enable this feature.
Apex REST API	✓			
Higher Limit for Concurrently Scheduled Apex Classes	✓			
Isolation of Test Data in Unit Tests	✓			
New IsTest (SeeAllData=true) Annotation	✓			

See Also:

[Summary of Spring '12 Features and Impact on Database.com Users](#)

API Enhancements

Feature	Automatically visible to all users. No setup required.	Automatically visible to all administrators. No setup required.	Not automatically visible. Feature is available but requires some setup.	Contact salesforce.com to enable this feature.
API Enhancements	✓			
Paging Using the <code>OFFSET</code> Clause				✓

See Also:

[Summary of Spring '12 Features and Impact on Database.com Users](#)

Additional Database.com Enhancements

Feature	Automatically visible to all users. No setup required.	Automatically visible to all administrators. No setup required.	Not automatically visible. Feature is available but requires some setup.	Contact salesforce.com to enable this feature.
Sandbox Terminology Changes	✓			
Field Updates for Encrypted Custom Fields	✓			
Workflow Field Updates Can Retrigger Workflow Rules	✓			
Test Database Retention Policy		✓		

See Also:

[Summary of Spring '12 Features and Impact on Database.com Users](#)

Database.com Console Enhancements

Database.com Console Updates

In Spring '12, the **User** module has been renamed to **Most Used Licenses**, but the functionality remains the same. Also under the **Business Logic** module, Workflow Rules has been renamed to Rules, and the functionality remains the same for Database.com.

The layout of the console has been updated, and bar graphs have been added to help monitor your usage. The functionality of the Database.com Console remains the same.

See Also:

[Schema Builder Enhancements—Beta](#)

[Platform Development Tools](#)

Schema Builder Enhancements—Beta



You asked for it! This enhancement is from an [idea](#) on the IdeaExchange.



Note: This release contains a beta version of schema builder that is production-quality but has known limitations. Schema builder is not supported on Microsoft® Internet Explorer® version 6 or earlier.

New in Spring '12, you can now create new custom objects, custom fields, and relationships with schema builder. This eliminates the need to click from page to page to find the details of a master-detail relationship or to add a new custom field to an object in your schema. For example, if you're using schema builder to view the details of your schema, you can add a new custom object without leaving schema builder. The drag-and-drop interface lets you easily add a custom object or new field, and saves the layout of your schema any time you move an object.

Schema builder is enabled by default and lets you add the following to your schema:

- Custom Objects
- Lookup Relationships
- Master-Detail Relationships
- Fields of the following types:
 - ◇ Checkbox
 - ◇ Currency
 - ◇ Date
 - ◇ Date/Time
 - ◇ Email
 - ◇ Number
 - ◇ Percent
 - ◇ Phone
 - ◇ Text

- ◇ Text Area
- ◇ Long Text Area
- ◇ Rich Text Area
- ◇ URL

See Also:

[Database.com Console Enhancements](#)
[Working with Schema Builder](#)

Working with Schema Builder



Note: This release contains a beta version of schema builder that is production-quality but has known limitations. Schema builder is not supported on Microsoft® Internet Explorer® version 6 or earlier.

To access schema builder, click **Schema Builder** in the Database.com console.

When working with schema builder:

- Click on an object and move it to any space on the canvas. Schema builder saves the layout of your schema any time you move an object.
- Click **Auto-Layout** to automatically sort the layout of the objects in your schema.







Important: Once you click **Auto-Layout**, you can't undo it.

- Click **View Options** to:
 - ◇ **Display Element Names** if you prefer system names, or **Display Element Labels** if you prefer text values.
 - ◇ **Show/Hide Relationships**
 - ◇ **Show/Hide Legend**
- The **Elements** tab lets you drag and drop new custom objects and fields onto the canvas.
 - ◇ To create a new custom object, see [Creating Objects with Schema Builder](#).
 - ◇ To create a new custom field, see [Creating Fields with Schema Builder](#).
- The **Objects** tab lets you select objects to display on the canvas.
 - ◇ Click the drop-down list in the sidebar to filter your list of objects:
 - All Objects
 - Selected Objects
 - Custom Objects
 - System Objects





Note: Objects created outside of schema builder, such as through an app or the API, don't automatically display on the canvas. Select the checkbox for the object created outside schema builder to display it on the canvas.

- ◇ To search for an object, type its name in the **Quick Find...** box.
- ◇ Hover over an object in your list of objects and click  to find it on the canvas.
- Hover over relationship lines to show relationship details such as lookup and master-detail relationships. Click the name of the object to find it on the canvas. You may want to hide relationships if your schema is taking too long to load.
- To view the details of a field in a new window, click the element name or label.
- Click  to:
 - ◇ **Hide Object on Canvas**
 - ◇ **View Object** detail in a new window
- Click **Show More Fields** to display all the fields of an object.
- To zoom in, click . To zoom out, click .



Note: You can't save the level of zoom when closing schema builder.

- To collapse the sidebar, click . To expand it, click .
- The map in the lower right corner shows the overall layout of your objects on the canvas. Click on the map to navigate the layout of your objects.
- To close the schema builder and save the layout of your objects, click **Close**.



Important: If your schema contains many objects and fields, it may cause long loading times. Click **Hide Relationships** to improve schema builder performance.

Creating Objects with Schema Builder



Note: This release contains a beta version of schema builder that is production-quality but has known limitations. Schema builder is not supported on Microsoft® Internet Explorer® version 6 or earlier.

To create a new custom object with schema builder:

1. Click the **Elements** tab.
2. Click **Object** and drag it onto the canvas.
3. Enter information to define your object. See “Schema Builder Custom Object Definition” in the online help for a list of object definitions.
4. Click **Save**.

Creating Fields with Schema Builder



Note: This release contains a beta version of schema builder that is production-quality but has known limitations. Schema builder is not supported on Microsoft® Internet Explorer® version 6 or earlier.

To create a new custom field with schema builder:

1. Click the **Elements** tab.
2. Click a field and drag it onto an object on the canvas. See “Schema Builder Custom Field Definition” in the online help for a list of available fields.
3. Enter a `Field Label`.

The `Field Name` is automatically populated based on the field label you enter. This name can contain only underscores and alphanumeric characters, and must be unique in your organization. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.

4. Enter a `Description` of the custom field.
5. Enter `Help Text` to detail the purpose and function of a custom field.
6. Enter a `Default Value` to automatically insert a value of a custom field when a new record is created.
7. Depending on the custom field type you choose, enter any remaining field attributes.
8. Click **Save**.

Working with Schema Builder



Note: This release contains a beta version of schema builder that is production-quality but has known limitations. Schema builder is not supported on Microsoft® Internet Explorer® version 6 or earlier.

To access schema builder, click **Schema Builder** in the Database.com console.

When working with schema builder:

- Click on an object and move it to any space on the canvas. Schema builder saves the layout of your schema any time you move an object.
- Click **Auto-Layout** to automatically sort the layout of the objects in your schema.







Important: Once you click **Auto-Layout**, you can't undo it.

- Click **View Options** to:
 - ◊ **Display Element Names** if you prefer system names, or **Display Element Labels** if you prefer text values.
 - ◊ **Show/Hide Relationships**
 - ◊ **Show/Hide Legend**
- The **Elements** tab lets you drag and drop new custom objects and fields onto the canvas.
 - ◊ To create a new custom object, see [Creating Objects with Schema Builder](#).
 - ◊ To create a new custom field, see [Creating Fields with Schema Builder](#).
- The **Objects** tab lets you select objects to display on the canvas.
 - ◊ Click the drop-down list in the sidebar to filter your list of objects:
 - All Objects
 - Selected Objects
 - Custom Objects
 - System Objects





Note: Objects created outside of schema builder, such as through an app or the API, don't automatically display on the canvas. Select the checkbox for the object created outside schema builder to display it on the canvas.

- ◇ To search for an object, type its name in the **Quick Find...** box.
- ◇ Hover over an object in your list of objects and click  to find it on the canvas.
- Hover over relationship lines to show relationship details such as lookup and master-detail relationships. Click the name of the object to find it on the canvas. You may want to hide relationships if your schema is taking too long to load.
- To view the details of a field in a new window, click the element name or label.
- Click  to:
 - ◇ **Hide Object on Canvas**
 - ◇ **View Object** detail in a new window
- Click **Show More Fields** to display all the fields of an object.
- To zoom in, click . To zoom out, click .



Note: You can't save the level of zoom when closing schema builder.

- To collapse the sidebar, click . To expand it, click .
- The map in the lower right corner shows the overall layout of your objects on the canvas. Click on the map to navigate the layout of your objects.
- To close the schema builder and save the layout of your objects, click **Close**.



Important: If your schema contains many objects and fields, it may cause long loading times. Click **Hide Relationships** to improve schema builder performance.

See Also:

[Schema Builder Enhancements—Beta](#)

Platform Development Tools

Developer Console—Generally Available

The Developer Console has a few changes and improvements in Spring '12. Most importantly, it is now opened by clicking **Your Name > Developer Console**. (Previously the menu item was **System Log**.)

Additional enhancements to the Developer Console in Spring '12 include:

Raw Debug Log Viewer

You can now directly view a “raw” log from the **Logs** tab of the Developer Console. Select a debug log and click **Open Raw Log**.

Previously if you wanted to view the raw unformatted version of a debug log in the Developer Console you had to switch back to the Old System Log. With this change, all features of the Old System Log are available in the Developer Console, and the Old System Log interface has been removed.

Heap Search

You can now search the contents of the heap in the new **Search** tab of the Heap Dump Inspector. Enter a value or heap address to search for and find all occurrences throughout the heap. Search matches partial symbol values, but addresses must be exact.

You can quickly search for a value in the heap by clicking the search icon (🔍) that appears to the right of the value when you hover over it in the State panel.

General Development Improvements

We've made a few changes that should improve your workflow while developing using the Developer Console. Most significantly, when validating your modified source views, all modified sources are validated together instead of individually. Changes that may be inconsistent with code on the server, but are consistent when taken together—such as what occurs during refactoring, and other common development tasks—will no longer report errors.

Also, if you save these changes using **Save All** they will be saved in one request, so you can save a collection of changes that would otherwise be blocked because separately they would be invalid.

To reflect this change in how validation is performed, the **Problems** panel is now globally available, allowing you to see compilation errors and other problems for all open source views.

The Developer Console will notice if a source view has been changed by another user since you opened it. If you haven't made any changes to it, the source view will be updated automatically. If you've made modifications, you'll see an alert that lets you know another user has made changes, with the option to update the source view to the latest version.



Caution: When you update to the latest version your changes will be overwritten. Copy your version out of the source view to preserve it. You can't save a modified source view if it has also been modified on the server. Instead, update to the latest version and integrate your modifications into the new version.

Finally, some simple workspace improvements:

- The Developer Console will now remember the open and collapsed state of various panels in the workspace, such as the Stack Trace, Source, and Execution Overview panels. This should make it easier and more worthwhile for you to configure a workspace that feels comfortable.
- The Developer Console will remember the line number you were on in source views, and remember your last executed code snippet in the Execute Anonymous box.

Force.com IDE—Generally Available

The Spring '12 release of the Force.com IDE adds updates for the latest Force.com platform features, and is available as a downloadable installer or through an Eclipse update site. The Spring '12 release of the Force.com IDE contains the following enhancements:

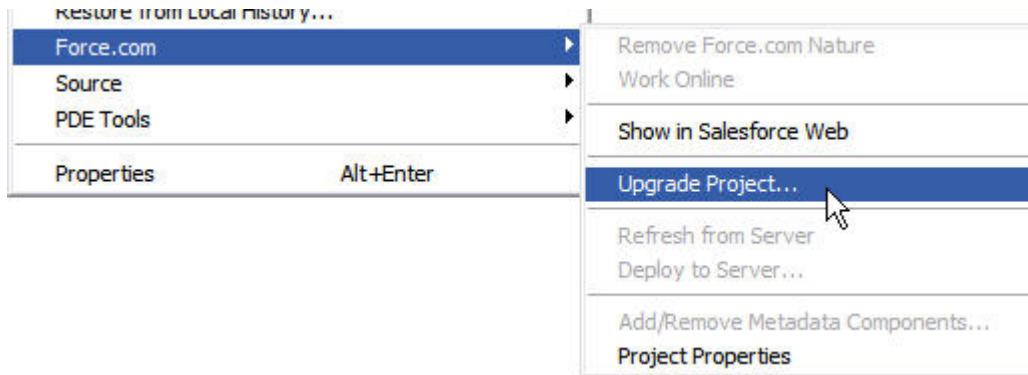
- Support for Metadata API version 24.0.
- We've added a number of new developer guides to the built-in documentation, including:
 - ◇ REST API Developer's Guide
 - ◇ Chatter REST API Developer's Guide
 - ◇ Streaming API Developer's Guide
 - ◇ Object Reference Guide
- All documentation has been updated with Spring '12 content.

To install the latest version of the Force.com IDE standalone application, download the appropriate installer from wiki.developerforce.com/page/Force.com_IDE_Installation.

To update the Force.com IDE Eclipse plugin, follow the directions available online at wiki.developerforce.com/page/Updating_the_Force.com_IDE.

After upgrading to the Spring '12 plugin, projects created in a Winter '09 or later Force.com IDE can be upgraded to Spring '12 using the Project Upgrade wizard. To upgrade a Force.com project:

1. Right click a project and choose **Force.com > Upgrade Project** to open the Project Upgrade wizard.



2. On the first page of the wizard, review the information and click **Next** to continue.
3. On the second page of the wizard, review the full details of what will be changed. If you don't want to upgrade all of these components, click **Cancel**. Otherwise click **Finish**.
4. On the final page of the wizard, review your changes.
5. Click **Finish** to retrieve the specified components.



Note: Force.com IDE releases prior to Winter '09 were non-upgradeable “Developer Preview” releases. To migrate to the latest Force.com IDE from a Developer Preview release, delete all Force.com projects from your workspace and uninstall the older IDE plugin from Eclipse, then install the latest release.

See Also:

[Database.com Console Enhancements](#)

Security Enhancements

Spring '12 introduces several new features that enhance security.

Single Sign-On Changes

Now when you save the Single Sign-On Settings page in a test database organization, if the Database.com Login URL value was for a single instance, the value is reformatted. The reformatting will only happen once, but it will occur even if you made no changes to the page. Be sure to review the Database.com Login URL value after you've saved the Single Sign-On Settings page.

New Password Complexity Option

The new `Must mix alpha, numeric, and special characters` password complexity option requires that the user must include at least one alphabetic character, one number, and one of the following characters ! # \$ % - _ = + < > in their password. Set the option from **Security Controls > Password Policies**.

Increased Maximum Timeout Value

The maximum session timeout value has increased from 8 to 12 hours. This value is the amount of time that passes before a user is logged out due to inactivity.

Requiring a Secure Connection (HTTPS) Is Permanent

You can no longer disable HTTPS access to an organization. If you currently have the `Require secure connections (HTTPS)` setting disabled, and then enable it, you can only switch back to HTTP by contacting your salesforce.com representative. This option is set from **Security Controls > Session Settings**.

Increased Service Provider Selection Availability

You can now set Service Providers from the Identity Provider page from any profile except sites and sites guest profiles. Service Providers are set from **Security Controls > Identity Provider**.

Encrypted Fields Available in Staging Database

Starting in Spring '12, a staging database can include the encryption keys for encrypted fields. This means that if there are encrypted fields in your production organization, those fields are also available for viewing and use in a staging database. Contact your salesforce.com representative to enable this feature.



Important: With this feature enabled, all users having a profile with “View Encrypted Data” in the staging database can see the actual value of any encrypted field.

Network Access Column Changes

The obsolete ISP, Organization, and Geography columns have been removed from the page at **Security Controls > Network Access**.

Sharing Enhancements

Name Fields for Groups, Queues, and Roles

When you create a group, queue, or role, you'll add a label and entity name for enhanced API access to the object. The label is the entity label as it appears on the user interface. The entity name is a unique name used to support API access to the object. You can customize the label and entity name from the UI or the Web Services API.

Default values are set for the entity name (for example, Group Name or Queue Name) in existing entities, sometimes appended with numbers to ensure that API Name is unique.

1. **Label.** The name of the entity as it appears on the UI. This value remains the same for existing entities. The Label field was previously known as the Name field.
2. **[Entity] Name.** The API Name corresponding to the `DeveloperName` field.

Name Fields for Sharing Rules

User Permissions Needed	
To create sharing rules:	"Manage Users"



You asked for it! This enhancement is from an [idea](#) on the IdeaExchange.

When you create a sharing rule, you'll add a label and rule name. The label is the sharing rule label as it appears on the user interface. The rule name is a unique name used to support API access to the object. For existing sharing rules, default values are set for the rule name, sometimes appended with numbers to ensure that API Name is unique. To create or edit a sharing rule: Click **Security Controls** > **Sharing Settings**. In the Sharing Rules section, click **New** or **Edit** next to the sharing rule you want to edit. Enter or change the sharing rule label and name.

1. **Label.** The name of the sharing rule as it appears on the UI. This value remains the same for existing sharing rules. The Label field was previously known as the Name field.

2. **Rule Name.** The API Name corresponding to the `DeveloperName` field.

Sharing Rule Components for Change Sets

Sharing rules can now be included as components in change sets, enabling you to send and receive sharing rule customizations between organizations. The components are: custom object criteria based sharing rule and custom object owner sharing rule.

Defer Sharing Calculations

User Permissions Needed	
To defer sharing calculations:	“Manage Users” AND “View Setup and Configuration” AND “Manage Sharing Calculation Deferral”
To suspend, resume or recalculate sharing rules:	“Manage Users”
To suspend and resume group membership calculation:	“Manage Users”

In Spring '12, when you make changes to roles or groups, you can suspend automatic group membership and sharing rule calculation. Performing a large number of configuration changes can lead to very long sharing rule evaluations or time outs. To avoid these issues, you can suspend these calculations and resume during your organization's maintenance periods.

Defer sharing calculation is ideal if you make a large number of changes to roles, groups, users, or public groups participating in sharing rules, and you want to suspend the automatic sharing calculation to a later time.



Note: The defer sharing calculation feature isn't enabled by default. To enable it for your organization, contact salesforce.com.

Group membership and sharing rule calculation are enabled by default.

If	You can
Group membership and sharing rule calculation are enabled	<ul style="list-style-type: none"> Suspend, update, and resume group membership calculation. This suspends sharing rule calculation and requires a full recalculation of sharing rules. Suspend, update, and resume sharing rule calculation.
Group membership calculation is enabled and sharing rule calculation is suspended	Suspend, update, and, resume group membership calculation.
Group membership calculation is suspended and sharing rule calculation is enabled	Suspend, update, resume, and recalculate sharing rule calculation.

To suspend or resume group membership calculation:

1. Click **Security Controls > Defer Sharing Calculations**.
2. In the Group Membership Calculations related list, click **Suspend**.



Note: If sharing rule calculations are enabled, suspending group membership calculations also suspends sharing rule calculations. Resuming group membership calculations also requires full sharing rule recalculation.

Users can't join a Chatter group during recalculation because adding a user to a Chatter group affects group membership tables, and group membership tables can't be changed during recalculation. Users can join a Chatter group when group membership recalculation is finished.

3. Make your changes to roles, groups, or users.
4. To enable group membership calculation, click **Resume**.

To suspend, resume, or recalculate sharing rule calculation:

1. Click **Security Controls > Defer Sharing Calculations**.
2. In the Sharing Rule Calculations related list, click **Suspend**.
3. Make changes to sharing rules, roles, or public groups participating in sharing rules.



Note: Any changes to sharing rules require a full recalculation.

4. To enable sharing rule calculation, click **Resume**.
5. To manually recalculate sharing rules, click **Recalculate**.

When sharing is recalculated, Database.com also runs all Apex sharing recalculations.

Apex Code Enhancements

Apex includes the following enhancements in Spring '12. Refer to the [Database.com Apex Code Developer's Guide](#) for complete information about Apex

The following enhancements are generally available:

Apex REST API

You can use Apex REST API to implement custom Web services in Apex and expose them through the REST API architecture.

The following changes have been made to Apex REST API:

- Apex REST automatically provides the REST request and response in your Apex REST methods via a static `RestContext` object. You no longer need to declare a `RestRequest` or `RestResponse` parameter in your method.
- User-defined types are now allowed as Apex REST parameter types.
- Order of elements in the JSON or XML response data no longer has to match the Apex REST method parameter order.

See the [Database.com Apex Code Developer's Guide](#) for more details about the Apex REST API.

Higher Limit for Concurrently Scheduled Apex Classes



You asked for it! The limit increase for concurrently scheduled classes is from an [idea](#) on the IdeaExchange.

The limit for the total number of Apex classes that can be scheduled concurrently increased from 10 to 25. With this higher limit, you can now schedule more Apex jobs for execution.

Isolation of Test Data in Unit Tests

Starting with Database.com API version 24.0, test methods don't have access to pre-existing data in the organization, such as custom objects and custom settings data. They can only access data that they create. This lets you to write more robust unit tests by preventing any dependencies of tests on any existing data in the organization.

Objects that are used to manage your organization or metadata objects can still be accessed in your tests, such as:

- User
- Profile
- Organization
- ApexClass
- ApexTrigger

This access restriction to test data applies to all code running in test context. For example, if a test method causes a trigger to execute and the test can't access organization data, the trigger won't be able to either.

Test code saved against Database.com API version 23.0 or earlier continues to have access to all data in the organization and its data access is unchanged.

You can choose to open up access of a test class or a test method to enable access to all data in the organization by using the `IsTest(SeeAllData=true)` annotation. See [New IsTest\(SeeAllData=true\) Annotation](#).

New IsTest(SeeAllData=true) Annotation

For Apex code saved using Database.com API version 24.0 and later, use the `isTest(SeeAllData=true)` annotation to grant test classes and individual test methods access to all data in the organization, including pre-existing data that the test

didn't create. Starting with Apex code saved using Database.com API version 24.0, test methods don't have access by default to pre-existing data in the organization. However, test code saved against Database.com API version 23.0 or earlier continues to have access to all data in the organization and its data access is unchanged. See [Isolation of Test Data in Unit Tests](#).

If a test class is defined with the `isTest(SeeAllData=true)` annotation, this annotation applies to all its test methods whether the test methods are defined with the `@isTest` annotation or the `testmethod` keyword.

This example shows how to define a test class with the `isTest(SeeAllData=true)` annotation. All the test methods in this class have access to all data in the organization.

```
// All test methods in this class can access all data.
@isTest(SeeAllData=true)
public class TestDataAccessClass {

    // This test accesses an existing merchandise item.
    // It also creates and accesses a new test merchandise item.
    static testmethod void myTestMethod1() {
        // Query an existing merchandise item in the organization.
        Merchandise__c m = [SELECT Id, Price__c, Total_Inventory__c, Description__c
                           FROM Merchandise__c WHERE Name='Pencils' LIMIT 1];
        System.assert(m != null);

        // Create a test merchandise item based on the queried merchandise item.
        Merchandise__c testMerchandise = m.clone();
        testMerchandise.Name = 'Test Pencil';
        insert testMerchandise;

        // Query the test merchandise that was inserted.
        Merchandise__c testMerchandise2 = [SELECT Id, Price__c, Total_Inventory__c
                                           FROM Merchandise__c WHERE Name='Test Pencil' LIMIT 1];
        System.assert(testMerchandise2 != null);
    }

    // Like the previous method, this test method can also access all data
    // because the containing class is annotated with @isTest(SeeAllData=true).
    @isTest static void myTestMethod2() {
        // Can access all data in the organization.
    }
}
```

This second example shows how to apply the `isTest(SeeAllData=true)` annotation on a test method. Because the class that the test method is contained in isn't defined with this annotation, you have to apply this annotation on the test method to enable access to all data for that test method. The second test method doesn't have this annotation, so it can access only the data it creates in addition to objects that are used to manage your organization, such as users.

```
// This class contains test methods with different data access levels.
@isTest
private class ClassWithDifferentDataAccess {

    // Test method that has access to all data.
    @isTest(SeeAllData=true)
    static void testWithAllDataAccess() {
        // Can query all data in the organization.
    }

    // Test method that has access to only the data it creates
    // and organization setup and metadata objects.
    @isTest static void testWithOwnDataAccess() {
        // This method can still access the User object.
        // This query returns the first user object.
        User u = [SELECT UserName,Email FROM User LIMIT 1];
    }
}
```

```

        System.debug('UserName: ' + u.UserName);
        System.debug('Email: ' + u.Email);

        // Can access the test invoice that is created here.
        Invoice_Statement__c inv = new Invoice_Statement__c(
            Description__c='Invoice 1');
        insert inv;
        // Access the invoice that was just created.
        Invoice_Statement__c insertedInv = [SELECT Id,Description__C
            FROM Invoice_Statement__c
            WHERE Description__c='Invoice 1'];
        System.assert(insertedInv != null);
    }
}

```

New isPermissionable Method for the Describe Field Result

In `Schema.DescribeFieldResult`, the `isPermissionable` method has been added. This method indicates whether field permissions can be specified for the field (`true`), or not (`false`).

Single Sign-On when Using Authentication Providers

Database.com provides the ability to use an authentication provider, such as Facebook[®] or Janrain[®], for single sign-on into Database.com. To set up single sign-on, you must create a class that implements `Auth.RegistrationHandler`. Classes implementing the `Auth.RegistrationHandler` interface are specified as the `Registration Handler` in authorization provider definitions, and enable single sign-on into Database.com organizations from third-party services such as Facebook. Using information from the authentication providers, your class must perform the logic of creating and updating user data as appropriate.

Name	Arguments	Return Type	Description
<code>createUser</code>	ID <i>portalId</i> Auth.UserData <i>userData</i>	User	Returns a User object using the specified portal ID and user information from the third party, such as the username and email address. The <i>portalID</i> value may be null or an empty key if there is no portal configured with this provider.
<code>updateUser</code>	ID <i>userId</i> ID <i>portalId</i> Auth.UserData <i>userData</i>	Void	Updates the specified user's information. This method is called if the user has logged in before with the authorization provider and then logs in again, or if your application is using the Existing User Linking URL. This URL is generated when you define your authentication provider. The <i>portalID</i> value may be null or an empty key if there is no portal configured with this provider.

The constructor for `Auth.UserData` has the following syntax:

```

Auth.UserData(String identifier,
               String firstName,
               String lastName,
               String fullName,
               String email,
               String link,
               String userName,
               String locale,

```

```
String provider,
String siteLoginUrl,
Map<String, String> attributeMap)
```

After a user is authenticated using an authentication provider, the access token associated with that provider for this user can be obtained in Apex using the `Auth.AuthToken` Apex class. `Auth.AuthToken` provides a single method, `getAccessToken`, to obtain this access token. For more information about authentication providers, see “About External Authentication Providers” in the Database.com online help.

Name	Arguments	Return Type	Description
<code>getAccessToken</code>	String <i>authProviderId</i> String <i>providerName</i>	String	Returns an access token for the current user using the specified 18-character identifier of an Auth. Provider definition in your organization and the name of the provider, such as Database.com or Facebook.

API Enhancements

Spring '12 (API version 24.0) improvements:

Changes across the API layer:

- [New and Changed Objects](#)
 - ◊ [Chatter API Objects](#)
- [SOQL Enhancements](#)

Changes to individual APIs:

- [SOAP Web Services API Enhancements](#)
- [REST API Enhancements](#)
- [Streaming API Enhancements](#)
- [Metadata API Enhancements](#)

See Also:

[New and Changed Objects](#)
[SOQL Enhancements](#)
[SOAP Web Services API Enhancements](#)
[REST API Enhancements](#)
[Streaming API Enhancements](#)
[Metadata API Enhancements](#)

New and Changed Objects

Generally Available Enhancements

For information about new and changed Chatter objects, see [Chatter API Objects](#) on page 25.

New Objects

These objects have been added in API version 24.0.

- The UserRecordAccess object has been added to check users' access to a set of records.
- These objects provide access to additional permissions in permission sets.

Object	Description
FieldPermissions	Represents the enabled field permissions for the parent PermissionSet.
ObjectPermissions	Represents the enabled object permissions for the parent PermissionSet.

- These objects let you track items related to Chatter Answers:

Object	Description
QuestionReportAbuse	Represents a user-reported abuse on a Question in a Chatter Answers community.
QuestionSubscription	Represents a subscription for a user following a Question.
ReplyReportAbuse	Represents a user-reported abuse on a Reply in a Chatter Answers community.

Changed Objects

The following objects have been changed in API version 24.0.

- The User object now includes the following fields to support Chatter:
 - ◊ UserPermissionsChatterAnswersUser, which indicates if a portal user has the Chatter Answers User feature license and is enabled to use Chatter Answers.
 - ◊ UserPreferencesDisableAllFeedsEmail
 - ◊ UserPreferencesDisableBookmarkEmail
 - ◊ UserPreferencesDisableChangeCommentEmail
 - ◊ UserPreferencesDisableFollowersEmail
 - ◊ UserPreferencesDisableLaterCommentEmail
 - ◊ UserPreferencesDisableLikeEmail
 - ◊ UserPreferencesDisableMentionsPostEmail
 - ◊ UserPreferencesDisableProfilePostEmail
 - ◊ UserPreferencesDisableSharePostEmail
 - ◊ UserPreferencesDisCommentAfterLikeEmail
 - ◊ UserPreferencesDisMentionsCommentEmail
 - ◊ UserPreferencesDisableMessageEmail
 - ◊ UserPreferencesDisProfPostCommentEmail
- These objects now include the DeveloperName and Name fields.
 - ◊ Group
 - ◊ UserRole
 - ◊ Queue
- The [CustomObject]__OwnerSharingRule object (where [CustomObject] represents the name of the custom object) now includes the DeveloperName and Name fields.

See Also:

[API Enhancements](#)

[Chatter API Objects](#)

Chatter API Objects

New Chatter Objects

These new objects have been added in API version 24.0.

Object	Description
QuestionSubscription	Represents a subscription for a user following a Question.

Changed Chatter Objects

These objects have been changed in API version 24.0.

- The ChatterActivity object, which represents Chatter activity statistics:
 - ◊ Now includes the number of likes received on comments in addition to likes received on posts.
 - ◊ No longer includes the Nillable property for the fields `CommentCount`, `CommentReceivedCount`, `LikeReceivedCount`, and `PostCount`.
- The ChatterMessage object now supports the `delete()` call and can be used to delete any user's Chatter messages; for example, for compliance purposes. After your organization is upgraded to Winter '12, ChatterMessage will support `delete()` in API version 23.0 as well as in version 24.0.
- Two new fields have been added to the FeedComment object so you can attach files to comments via the API:
 - ◊ `CommentType` is the type of comment. File attachments are of type `ContentComment`.
 - ◊ `RelatedRecordId` is the ID of the `ContentVersion` object associated with a `ContentComment`.



Note: Prior to API version 24.0, a text entry was required on a comment. As of version 24.0, a text entry is optional if the `CommentType` is `ContentComment`—an uploaded file on a comment. If you have any Apex triggers associated with the FeedComment object, keep in mind that this change may affect the behavior of existing triggers.

- The CollaborationGroupMember object:
 - ◊ Supports the `update()` call.
 - ◊ Includes the Nillable property for the `NotificationFrequency` field.
 - ◊ Includes the `CollaborationRole` field, which represents the role of a member in a group. Group owners and managers can assign group members of their groups to `Member` or `Manager` roles.

See Also:

[New and Changed Objects](#)

SOQL Enhancements

Use the Salesforce Object Query Language (SOQL) to construct query strings used in Database.com APIs. For a full description of the SOQL query syntax, see [Salesforce Object Query Language \(SOQL\)](#) in the *Web Services API Developer's Guide*.

Paging Using the OFFSET Clause — Developer Preview



Note: OFFSET is currently available as a Developer Preview. For more information on enabling OFFSET for your organization, contact Salesforce.com.

Use OFFSET to specify the starting row offset into the result set returned by your query. Using OFFSET is helpful for paging into large result sets, in scenarios where you need to quickly jump to a particular subset of the entire results. For example, the following SOQL query returns a result set that skips the first 100 rows of the full query results:

```
SELECT Name
FROM Merchandise__c
WHERE Price__c > 5.0
ORDER BY Name
LIMIT 50
OFFSET 100
```

See Also:

[API Enhancements](#)

SOAP Web Services API Enhancements

New and Changed Calls

Changed Calls

These calls have been changed in API version 24.0.

Call	Argument or Result Object	Field	Change	Description
describeSObjects()	DescribeSObjectResult	Field	permissionable property added	Indicates whether FieldPermissions can be specified for the field (<code>true</code>) or not (<code>false</code>).

Previous Versions

For links to documentation for previous versions of the API, see the What's New section of the relevant API or object reference document.

See Also:

[API Enhancements](#)

REST API Enhancements

The Force.com REST API leverages the simplified approach of REST to allow developers to more easily interact with other Web 2.0 applications such as Amazon.com AWS, Microsoft Azure, Google, Facebook, Twitter, and others:

- Interact with the application in a simplified way
- Integrate with third-party cloud systems and services
- Support mashup and Web 2.0 projects

If you need to move large amounts of data, use the Bulk API, which is built on RESTful principles, but is asynchronous and optimized for large data-loading tasks.

User Password Management

You can now securely access and manage user passwords using the new User Password methods in the Force.com REST API. For more information see the [Database.com REST API Developer's Guide](#).

See Also:

[API Enhancements](#)

Streaming API Enhancements

Streaming API—Generally Available

Streaming API is now generally available. Use Streaming API to receive notifications for changes to data that match a SOQL query you define, in a secure and scalable way.

Notification Enhancements

PushTopic records have been enhanced to provide more flexibility when specifying which events and data changes generate notifications. Two new PushTopic fields provide the flexibility:

- `NotifyForOperations` specifies which record events generate a notification. Valid field values are:
 - ◊ All (default)
 - ◊ Create
 - ◊ Update
- `NotifyForFields` specifies how a new or updated record is evaluated against the PushTopic query. Valid field values are:
 - ◊ All
 - ◊ Referenced (default)
 - ◊ Select
 - ◊ Where

For existing PushTopics, the `NotifyForOperations` will default to All and the `NotifyForFields` will default to Referenced. For more information see, [Event Notification Rules](#) in the Database.com Streaming API Developer's Guide.

Existing Streaming API PushTopics may behave quite differently in this release. In previous versions, Streaming API generated notifications when a record was created or updated as long as the record matched the PushTopic query. If the query had a WHERE clause, then a notification was generated when a record was created or updated, the record matched the PushTopic query, and one of the fields referenced in the WHERE clause was modified. Now record events that trigger a notification are specified by the `NotifyForOperations` field.

In previous versions, Streaming API only evaluated the fields referenced in the WHERE clause. However, now the API evaluates fields in the SELECT clause, the WHERE clause, or both depending on what you set the `NotifyForFields` field to.

Server Polling Frequency

Streaming API polls the server for activity and issues the resulting notifications to the channel. The polling frequency has decreased from five to three seconds. The actual time may fluctuate depending on the overall server load.

Maximum Request Size

The maximum size of the HTTP request post body that the server can accept from the client is 32,768 bytes, for example, when you call the CometD `subscribe` or `connect` methods. If the request message exceeds this size, the following error is returned in the response: 413 Maximum Request Size Exceeded. To keep requests within the size limit, avoid sending multiple messages in a single request.

See Also:

[API Enhancements](#)

Metadata API Enhancements

New Metadata Types

These metadata types are new in Metadata API version 24.0.

Metadata Type	Description
Group	Represents a public group, which can have users, roles, and other groups.
Queue	Represents a holding area for items before they are processed.
Role	Represents a user role.
SharingRules	Represents a set of sharing rules that's used to share records with a set of users, based on rules that specify the access level to the role or group. You can't create a SharingRules component directly. Use the <code>CustomObjectSharingRules</code> type, which extends it.

Updated Metadata

These metadata fields have been added or changed in Metadata API version 24.0.

Metadata Type or Related Object	Field or Field Type	Change	Description
Workflow	reevaluateOnChange	New	A new field on WorkflowFieldUpdate. When set to true, if the field update changes the field's value, all workflow rules on the associated object are re-evaluated. Any workflow rules whose criteria are met as a result of the field value change will be triggered.

See Also:[API Enhancements](#)

Additional Database.com Enhancements

Sandbox Terminology Changes

In Spring '12, there are some terminology changes for sandboxes but the underlying functionality is unchanged.

- A sandbox is now a test database.
- A developer sandbox is now a QA database.
- A full sandbox is now a staging database.

Field Updates for Encrypted Custom Fields

With Spring '12, we've expanded the reach of workflow field updates to include encrypted custom fields. Note, however, that if you try to use a formula to set an encrypted custom field's new value, the encrypted custom field isn't available in the formula editor as a resource.

Workflow Field Updates Can Retrigger Workflow Rules

We added a new checkbox option to workflow field updates that allows you to select whether you want a field update to trigger a re-evaluation of all workflow rules on the object.

Here's how it works:

- If the field update changes the field's value, all workflow rules on the associated object are re-evaluated. Any workflow rules whose criteria are met as a result of the field update will be triggered.
- If any of the triggered workflow rules result in another field update that's also enabled for workflow rule re-evaluation, a domino effect occurs, and more workflow rules can be re-evaluated as a result of the newly-triggered field update. This cascade of workflow rule re-evaluation and triggering can happen up to five times after the initial field update that started it.
- In a batch update, workflow is only retriggered on the entities where there is a change.
- Only workflow rules on the same object as the initial field update will be re-evaluated and triggered.
- Only workflow rules that didn't fire before will be retriggered.
- Cross-object workflow rules and time-based workflow rules aren't candidates for re-evaluation.
- Cross-object field updates that cause a field value to change don't trigger workflow rule re-evaluation on the associated object.

Test Database Retention Policy

In Spring '12, we've updated our test database retention policy to optimize our server capacity and support growth of our test database infrastructure. Capacity can affect the performance, backup, and replication of your test database instances. These policies help ensure that test databases use capacity more efficiently.

Unactivated Test Databases

New test databases that aren't activated within 30 days will be deleted. You'll get at least three notifications prior to scheduling the test database for deletion.

Locked Test Databases

Test databases that have been locked for 30 days will be deleted. You'll get at least three notifications prior to scheduling the test database for deletion. Test databases become locked when all the licenses for that type of test database expire.



Note: Deletion of a test database doesn't terminate or change any of your test database subscriptions. If you have a test database subscription and your test database is deleted, your subscription remains in effect and you can create a new test database.

Help and Training Enhancements

Tip Sheets and Implementation Guides

The following new or updated documents are now available:

- Two appendices have been added to the *ISVforce Guide*: one of which compares two ISVforce user licenses and the second of which compares three OEM user licenses.
- *Using Dashboard Filters*
- *Using Cross Filters in Reports*
- *What are Joined Reports?*
- *Using Bucket Fields*
- *Chatter Answers Implementation Guide*

Developer Guides

The following new or updated developer documents are now available:

- *Database.com SOQL/SOSL Reference*—The SOQL/SOSL Reference updated for Database.com.
- *Database.com Metadata API Developer's Guide*—The Metadata API Developer's Guide updated for Database.com.

Online Help

With Spring '12, Database.com help links now open an online help system that displays only help related to Database.com. Information about other salesforce.com products is no longer visible in this new help portal.