Database.com Summer '12 Release Notes

Last updated: April 16 2012

Table of Contents

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. Summer '12 includes some of your top ideas. Visit IdeaExchange for a complete list of ideas coming in Summer '12.

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

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

Summer '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.

General 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.
Browser Support Changes	✅			

Database.com Console Enhancements

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.
Workspaces		✅		
Navigating through Tab History		✅		
Navigating to Method and Variable Declarations		✅		
Syntax Highlighting		✅		
Performance Tree		✅		
Overlaying Apex Code and SOQL Statements		✅		

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.
Execute Anonymous Apex Code Enhancements		✓		

Schema Builder--Generally Available

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.
Edit properties of a custom field	✓			
Manage permissions for a custom field	✓			
Custom fields support all types	✓			

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.
Verifying the Expiration Date for a CA Certificate	✓			
Enable SMS-based Identity Confirmation			✓	
Support for Bearer Authentication Headers	✓			
Specify the HttpOnly Attribute		✓		

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.
Clickjacking Protection Applied to Setup Pages	✓			

Permission Sets 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.
Object and Field Permissions Page Renamed	✓			
Permission for Setting Apex Class Access		✓		

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 requires some setup.	Contactsalesforce.com to enable this feature.
Changed isUpdateable Method of DescribeFieldResult	✓			
DML Support for Permission Set sObjects	✓			
JSON Parsing Enhancements	✓			
Callout Limit Increase for Batch Apex	✓			
Apex Test Execution and Code Coverage Results Changes	✓			
New Type Methods	✓			

Feature	Automatically visible to all users. No setup required.	Automatically visible to all administrators. No setup required.	Not automatically visible. Feature is available requires some setup.	Contactsalesforce.com to enable this feature.
Change in Running Asynchronous Processes in Test Methods	✓			
Sorting Support for Non-Primitive Data Types in Lists	✓			
Active Query Cursor Limit Increase	✓			

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	✓			
SOQL OFFSET—Generally Available	✓			

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.
Enhanced Profile User Interface: Object and Tabs Page Renamed	✓			
Login Access Policy				✓
Allow Reparenting Option in Master-Detail Relationship Definitions		✓		
Object Limits Displayed on Custom Object Definitions		✓		

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.
Changes in Staging Test Database Configuration		✅		
Correction toTest Database Retention Policy		✅		
New Lookup Relationship Options	✅			✅ For one option
Formula Function Changes	✅			
Supported Locale Online Help Enhanced	✅			
Custom field labels changes	✅			
Time-Dependent Field Updates Can Retrigger Workflow Rules	✅			

General Enhancements

Summer ‘12 includes several general enhancements.

Browser Support Changes

Windows® Internet Explorer® version 6 is no longer supported. We recommend these supported browsers:

Browser	Comments
Microsoft® Internet Explorer® versions 7, 8, and 9	Salesforce.com recommends using Internet Explorer version 9. Apply all Microsoft hotfixes. The compatibility view feature in Internet Explorer 8 and 9 is not supported in Database.com. For configuration recommendations, see “” in the online help.
Mozilla® Firefox®, most recent stable version	Salesforce.com recommends using Firefox for best performance and makes every effort to test and support the most recent version. For configuration recommendations, see “” in the online help.
Google Chrome™, most recent stable version	Google Chrome applies updates automatically; Salesforce.com makes every effort to test and support the most recent version. There are no configuration recommendations for Chrome.
Apple® Safari® version 5.1.x	There are no configuration recommendations for Safari.

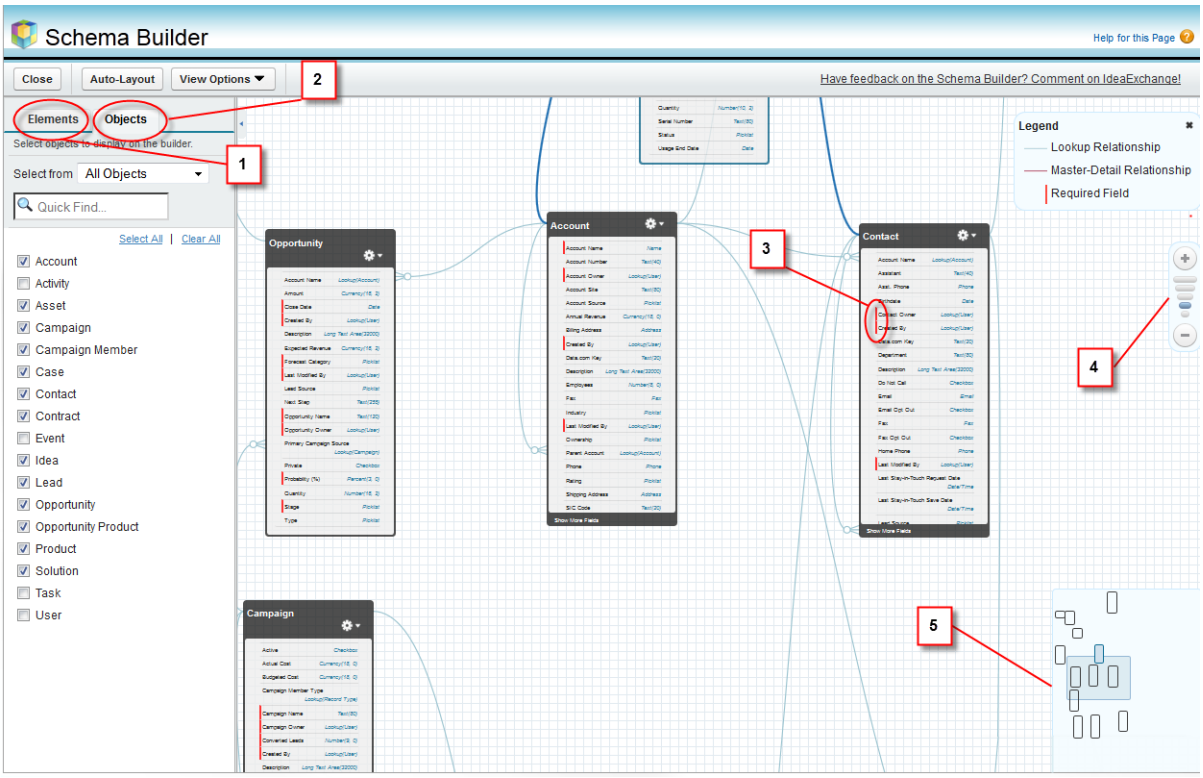
For information on feature-specific recommendations, see “Supported Browsers” in the online help.

DATABASE.COM CONSOLE ENHANCEMENTS

Schema Builder--Generally Available

With the Summer '12 release, the graphical Schema Builder is now generally available in all editions.

Schema Builder gives you a visual interface for developing database schemas. You can use it to create objects as well as custom fields of any type, including master-detail and lookup relationships, formulas, and roll-up summaries.



1. Use the Elements tab to create fields and objects by dragging them onto the canvas.
2. Use the Objects tab to find and show/hide objects.
3. A red line indicates the field is required.
4. Use the zoom control to view more or less of the schema.
5. The bird's eye view indicates where you are in the schema.

Summer '12 includes the following enhancements:

Edit properties of a custom field

We enhanced the Schema Builder documentation to describe the Edit Field Properties... option in the context menu for custom fields. To edit properties of a custom field, right-click the field name or label and select **Edit Field Properties...**.

Manage permissions for a custom field

To manage permissions of a custom field, click the element name or label and select **Manage Field Permissions**. Use the dialog box that appears to manage the field's visibility and writability for all standard and custom profiles.

Custom fields support all types

Schema Builder completes its support of custom field types by adding support for the following types:

Type	Description
Auto Number	Automatically assigns a unique number to each record. The maximum length of any auto-number field is 30 characters, 20 of which are reserved for prefix or suffix text.
Formula	Allows users to automatically calculate values based on other values or fields such as merge fields. See "Building Formulas" in the online help and "Operators and Functions" in the online help.  **Note:** Database.com uses the round half up tie-breaking rule for numbers in formula fields. For example, 12.345 becomes 12.35 and –12.345 becomes –12.34. In Database.com, the Formula editor does not provide a **Check Syntax** button. Syntax checking occurs when the user attempts to save the formula.
Picklist	Allows users to select a value from a list you define.
Picklist (Multi-select)	Allows users to select more than one picklist value from a list you define. These fields display each value separated by a semicolon.
Roll-Up Summary	Automatically displays the record count of related records or calculates the sum, minimum, or maximum value of related records. The records must be directly related to the selected record and on the detail side of a custom master-detail relationship with the object that contains the roll-up summary field. For example, a customfield called "Total Number of Guests" displays the number of guest custom object records in the Guests related list.
Text (Encrypted)	Allows users to enter any combination of letters, numbers, or symbols that are stored in encrypted form. You can set a maximum length of up to 175 characters. Encrypted fields are encrypted with 128-bit master keys and use the AES (Advanced Encryption Standard) algorithm. You can archive, delete, and import your master encryption key. To enable master encryption key management, contact

Type	Description
	salesforce.com. See also "Managing Master Encryption Keys" in the online help.

Developer Console Enhancements

Summer '12 includes several updates to the Developer Console. The following updates are generally available:

Creating and Deleting Workspaces in the Developer Console

A workspace is a collection of tabs (also called *views*) located in the lower panel of the Developer Console. A tab can contain a log, source code, a heap dump, or an object view.

You can create a workspace for each group of tabs that you use together. Then, while you're working in the Developer Console, switch between workspaces to keep your work organized. For example, the Apex Workbook creates and edits the classes KitchenUtility, Fridge, and Toaster. While working on the Apex Workbook, you could open these classes and save them as a workspace called *Apex Workbook Source*. You could then open debug logs for these classes and save them as a workspace called *Apex Workbook Test*. Switch between the workspaces as you code and test.

To create a workspace:

1. Click **Workspace:*Name***, then select **Workspace Manager**.
2. Click **New** and enter a name for the workspace.
3. Open the logs, source code, heap dumps or object views that you want grouped together in the workspace. When you switch to a different workspace or close the Developer Console, the Developer Console saves the state of the tabs (and the panels within the tabs) in the current workspace.

To delete a workspace:

1. Click **Workspace:*Name***, then select **Workspace Manager**.
2. Select a workspace and click **Delete**.

To rename a workspace:

1. Click **Workspace:*Name***, then select **Rename Current Workspace**.
2. Enter a new name and click **OK**.

Navigating through Tab History

When you open a log, a heap dump, or an item from the repository, it opens in a new tab (also called a *view*) in the workspace. You can have multiple tabs open at the same time.

To move backward and forward through your tab history, click the  buttons. You can also use these shortcuts:

- Backward—CTRL+,
- Forward—CTRL+.

Clicking  moves through your previously viewed tabs in the order that you viewed them. When you click the  button, the  button becomes active.

Navigating to Method and Variable Declarations

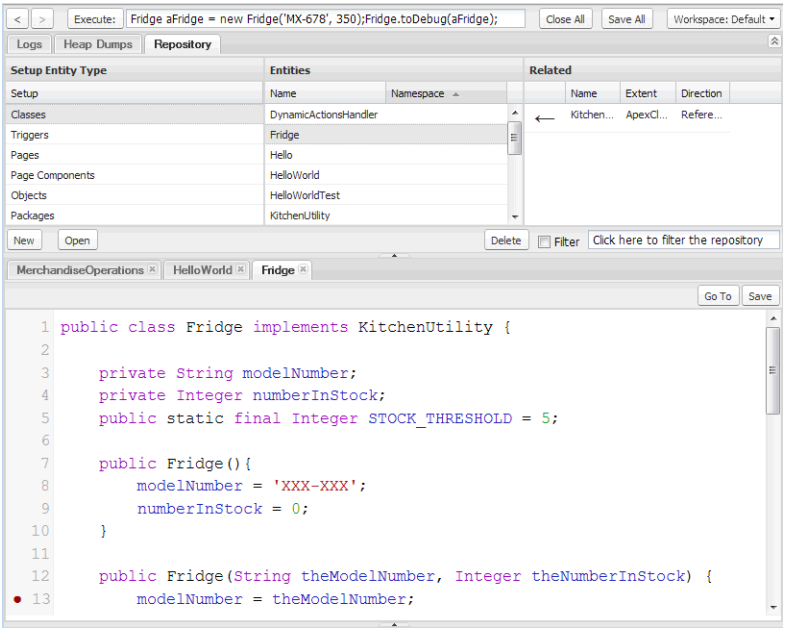
You can navigate directly to a method or variable declaration, rather than having to scroll or search to find it.

1. Click in a method or variable name.
2. Press CTRL+ALT+N or click **Go To** to move the cursor to the declaration.

   If the declaration is in another file, the file opens in a new tab.

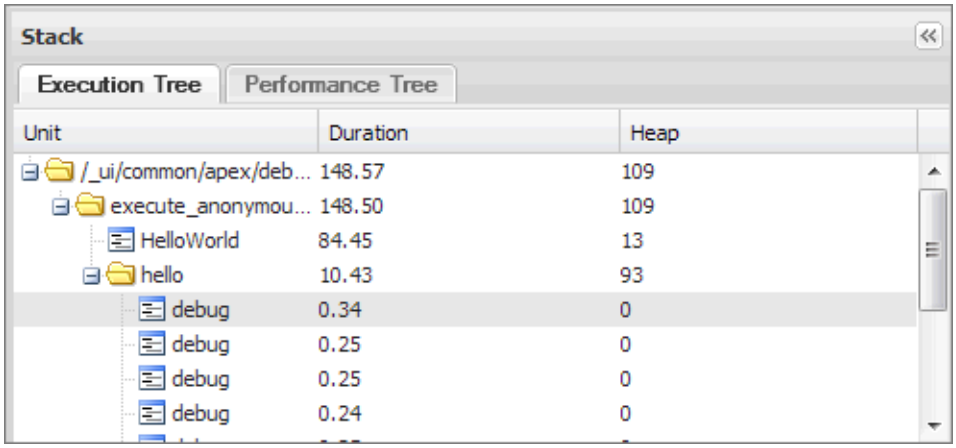
Syntax Highlighting

Syntax highlighting colors comments, numbers, strings, reserved keywords, primitive data types, variable declarations, and references.

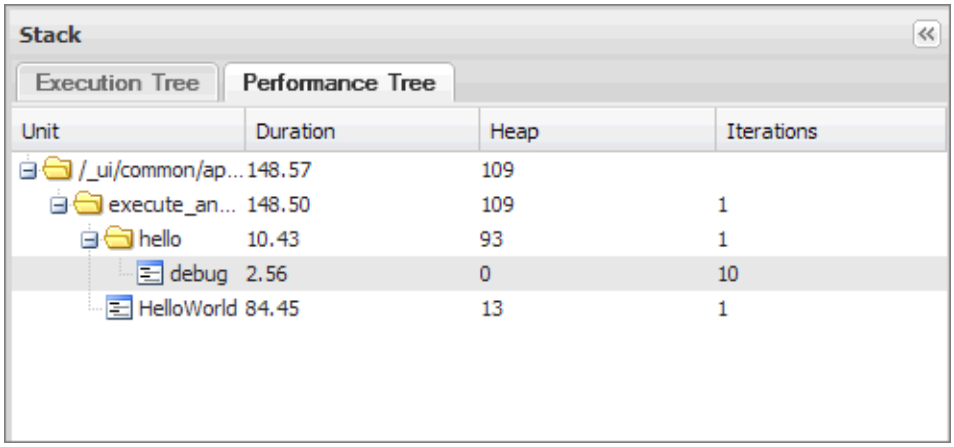


Performance Tree

In System Log view, the Stack pane now includes a performance tree. The performance tree aggregates operations to give you a better look at the performance of an operation as a whole. For example, if a `for` loop calls `System.debug()` 10 times, you see the duration of each call in the execution tree:



In the performance tree, the calls are aggregated so you see the total duration of every call to `debug` from `hello`:

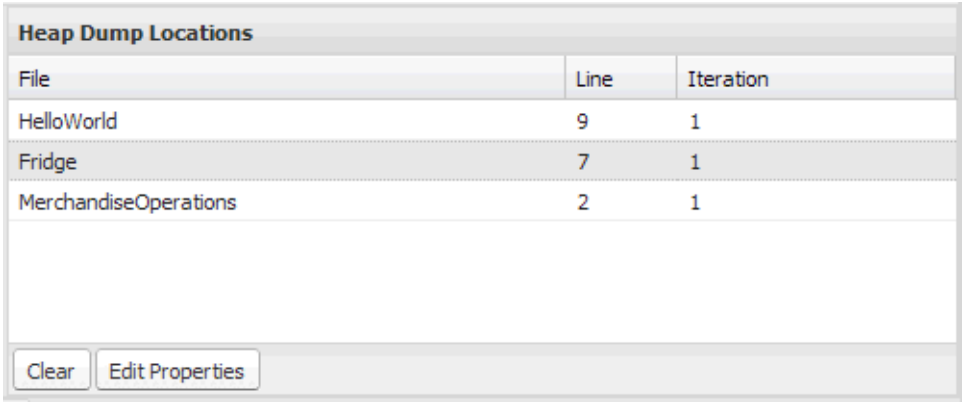


This example calls `debug` only from the `hello` method. Calls to `debug` from other locations are aggregated in those nodes. Calls from all locations are aggregated in the executed unit.

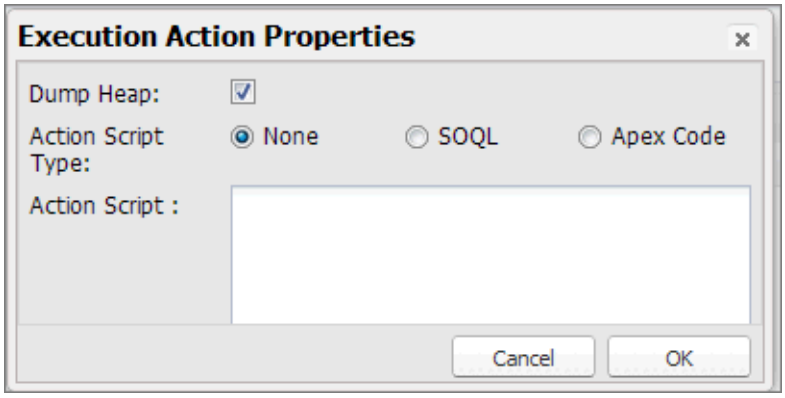
Overlaying Apex Code and SOQL Statements

When troubleshooting a runtime issue, you might want information about the state of a variable or the state of the database. You might also want to create a specific condition in which to test your code. Use the Developer Console to overlay a diagnostic output on the run without instrumenting the code. Write Apex code and SOQL statements that run when code at a heap dump location executes.

1. Select a heap dump location.



2. Click **Edit Properties**.
3. Select SOQL or Apex Code.

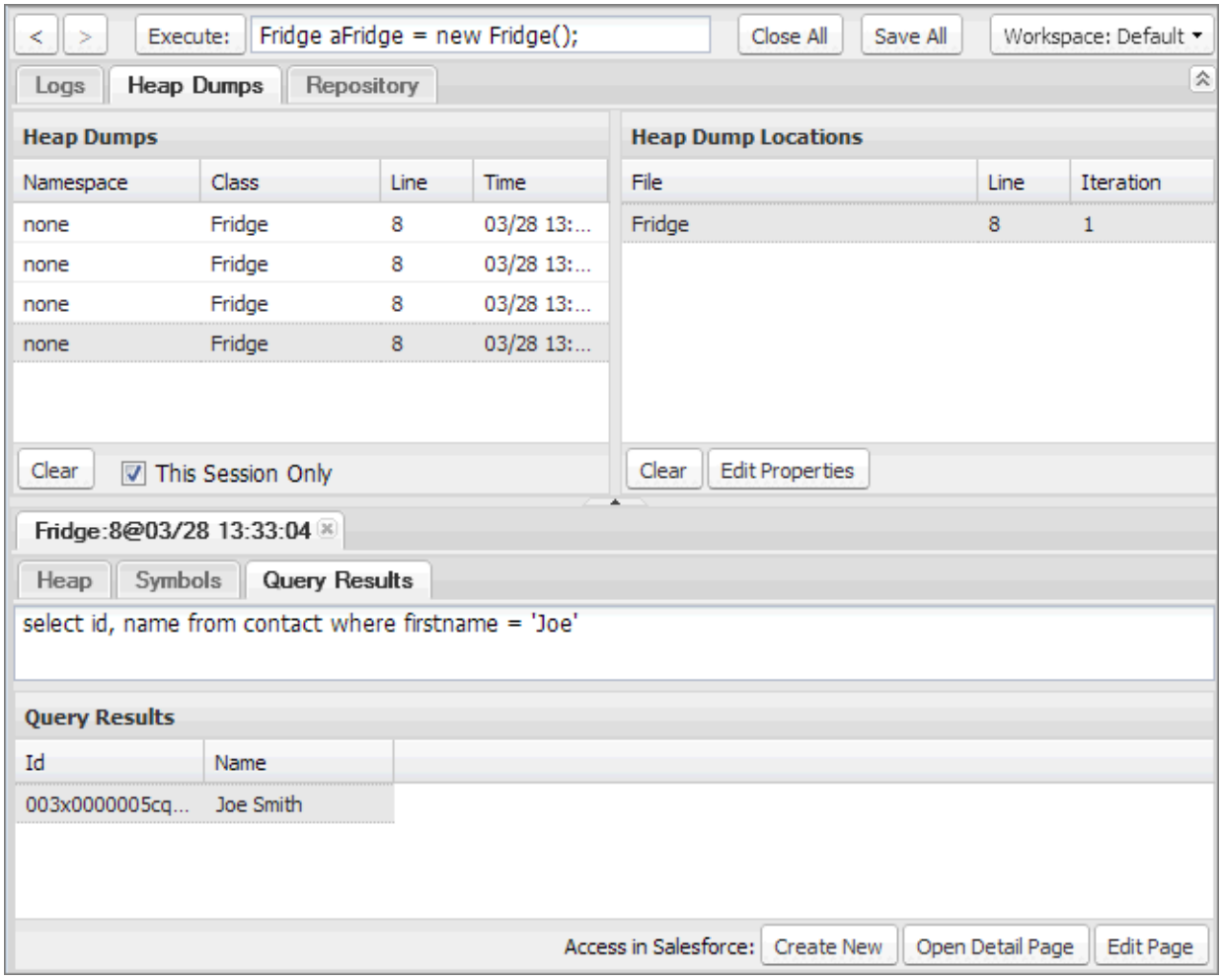


4. To run the diagnostic code without generating a heap dump, deselect **Dump Heap**.
5. Enter SOQL or Apex code in the script box and click **OK**.

Note:

You can't refer to local objects because an anonymous block is a new stack frame. Refer to static objects or create new objects. Also, you can't use bind variables in SOQL queries used in overlays.

View the results of a SOQL query in Query Results view:

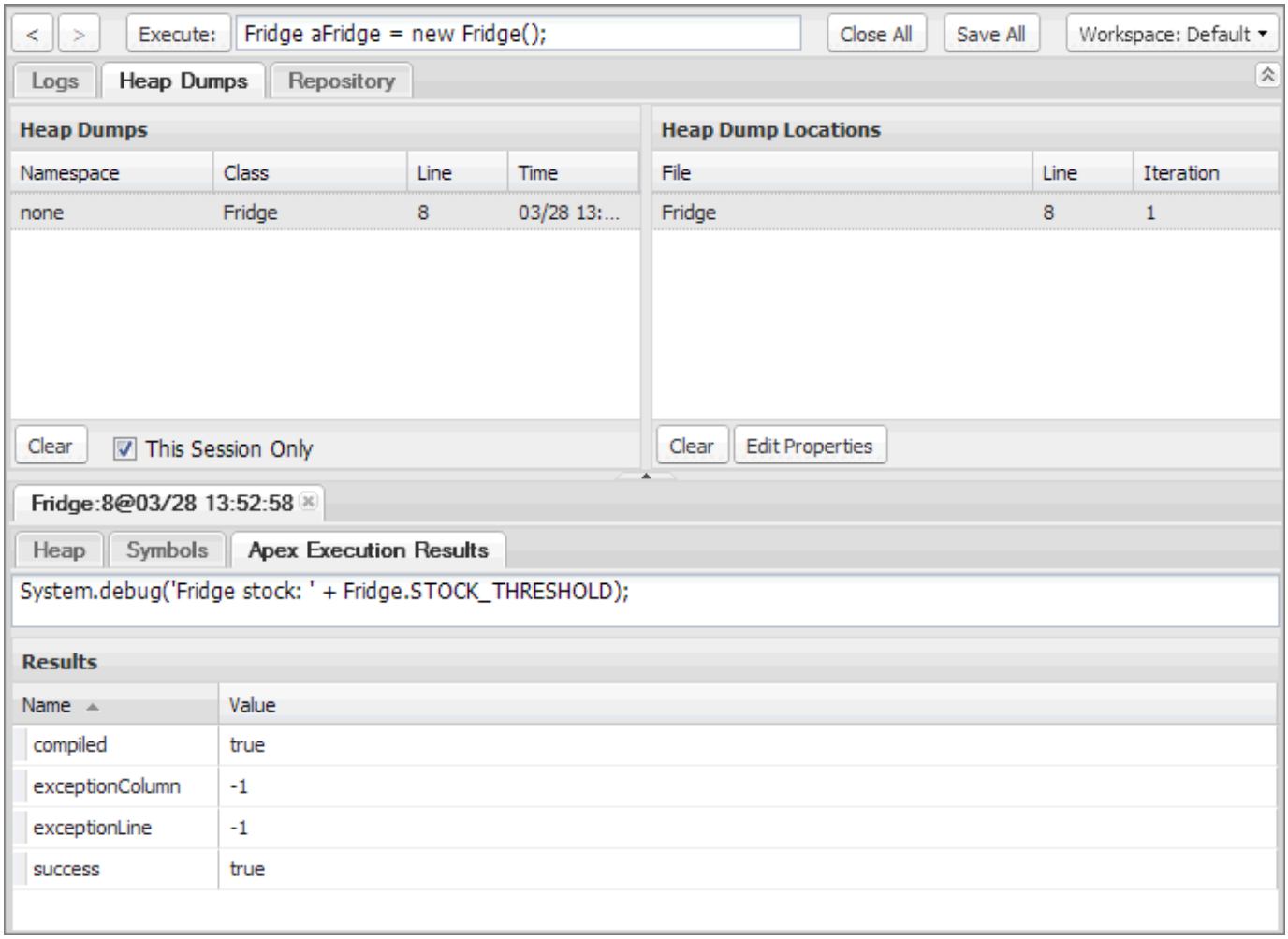


To create a new record for the outer entity of the `select` statement, click **Create New**.

To open a record's detail page in Salesforce, select the row and click **Open Detail Page**.

To edit a record in Salesforce, select the row and click **Edit Page**.

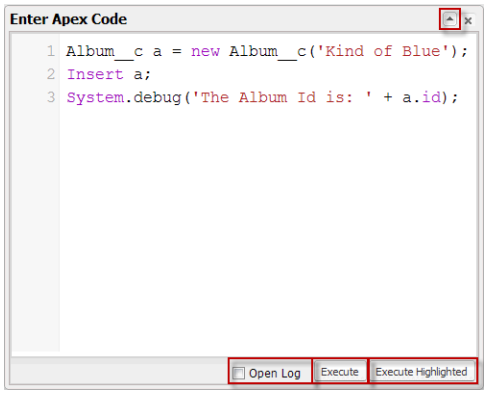
View Apex execution results in Apex Execution Results view:



Note: The value `-1` for `exceptionColumn` and `exceptionLine` indicates that these fields are not applicable.

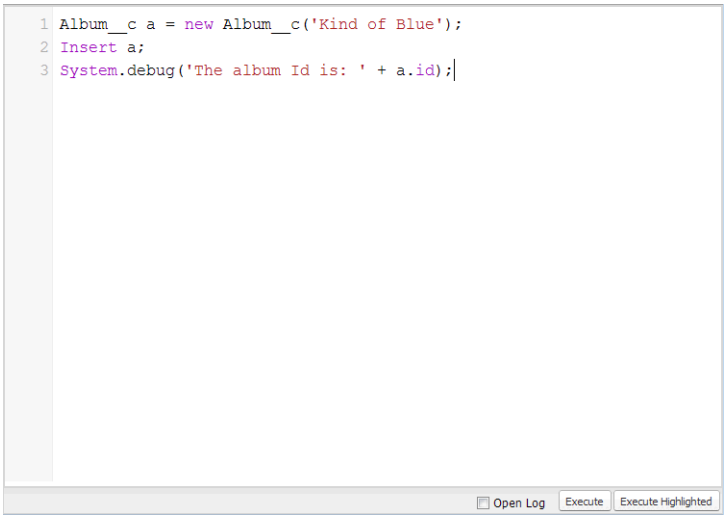
Execute Anonymous Apex Code Enhancements

The Summer '12 release includes several significant enhancements to the Execute Anonymous Apex code feature. The Execute text box has been moved to the top of the Developer Console so you can access it no matter which tool is selected. When you click in the Execute text box, an Enter Apex Code window pops up.



This window has several enhancements:

- To automatically open a debug log when the code executes, select **Open Log**.
- To execute a section of the code, select those lines and click **Execute Highlighted**. (To execute all the code in the window, click **Execute**, as you did in previous releases.)
- To open an execute anonymous editor in a new browser window, click  .



Security Enhancements

Summer ’12 introduces several new features that enhance security.

Verifying the Expiration Date for a CA Certificate

Prior to Summer ’12, the expiration date shown on the Certificate and Key Management page could be incorrect for certificates signed by an external certificate authority (CA). For example, you could have received a Certificate Signing Request through Database.com and then had it signed by a CA. Database.com could have set the certificate’s life at two years, but the CA may have set the certificate to be valid for one year. When you returned the signed certificate to Database.com for use, the original two-year expiration may not have been updated to one year in the user interface.

Verify a CA-signed certificate’s expiration date by clicking the certificate’s label in the list at **Security Controls** > **Certificate and Key Management**. On the Certificates page, the Expiration value in the Certificate entry is the correct date.

Enable SMS-based Identity Confirmation

SMS-based identity confirmation is now available for all users in addition to email-based confirmation. Database.com will use either of these methods to verify the user when necessary, such as when a user tries to log in from an unknown IP address.

Administrators enable this feature by checking `Enable SMS-based identity confirmation` at **Security Controls** > **Session Settings**.

Once the feature is enabled, every user sees a screen after logging in that asks them to register for mobile verification. The user can take one of the following actions:

- Enter a mobile phone number and then have it verified with a text message.
- Skip entering a mobile number now, but get asked again at their next login.
- Completely opt out of mobile verification.

Administrators can also enter a user’s mobile number, including country code, and pre-verify it. If `Enable SMS-based identity confirmation` is set when an administrator enters a mobile number for a user, or when a mobile number is set from an API using the `User` object, the mobile number is automatically considered verified. If `Enable SMS-based identity confirmation` is not set, the new mobile phone number is not considered verified. “Verified” means that Database.com will not ask the user to verify a mobile phone number at login, and that Database.com will use the number to send the user a verification code when necessary for SMS-based identity confirmation.

Support for Bearer Authentication Headers

OAuth access tokens may now be presented to the APIs using the OAuth bearer token format:

```
Authorization: Bearer access_token
```

For example, provide the access token in your REST requests:

```
curl https://instance_name.salesforce.com/services/data/v20.0/ -H 'Authorization: Bearer
 access_token'
```

The previous syntax, `Authorization: OAuth access_token`, is still supported.

Specify the HttpOnly Attribute

You can now specify the HttpOnly attribute for your session ID cookies. If the HttpOnly flag is included in the HTTP response header, the cookie cannot be accessed through client-side scripts. Set the HttpOnly attribute by selecting `Require HttpOnly attribute` at **Security Controls** > **Session Settings** for new and existing orgs.

Note: If you have a custom application that uses JavaScript to access session ID cookies, selecting `Require HttpOnly attribute` breaks your application because it denies the application access to the cookie. The Developer Console and AJAX Toolkit debugging window are also not available if the `Require HttpOnly attribute` is selected.

Clickjacking Protection Applied to Setup Pages

Setup pages now include protection against clickjack attacks, which are also known as user interface redress attacks. Setup pages are those pages accessed from the left side of the screen in the Developer Console. While there are reasons to frame pages, framed pages can be used by hackers.

If your organization displays Database.com user interface pages within a frame or iframe, then it's possible that setup pages will either display as a blank page or without the frame. The behavior varies depending on your browser and its version. Take one of the following actions to ensure that these pages will continue to work correctly in your organization:

- Don't turn off clickjacking protection for your organization. Instead, discontinue displaying these pages within a frame or iframe. This is the recommended solution.
- Ask Database.com support to disable clickjack protection for your organization. This allows you to continue framing Database.com pages, but the setup pages remain vulnerable to clickjack attacks. This solution is not recommended, but is available.

Permission Sets Enhancements

Object and Field Permissions Page Renamed

In Summer '12, the Object and Field Permissions page is now named "Object Settings." This page has the same functionality—only the name has changed.

To view this page, click **Manage Users** > **Permission Sets**, select a permission set, then click **Object Settings**.

Permission for Setting Apex Class Access

Starting in Summer '12, the "Customize Application" permission is no longer required to edit users' access to Apex classes. Now only the "Manage Users" permission is required. This change applies to profiles as well as permission sets.

Apex Code Enhancements

Apex includes the following enhancements in Summer '12. For detailed information on these enhancements, refer to the *Database.com Apex Code Developer's Guide*.

Changed isUpdateable Method of DescribeFieldResult

The `isUpdateable` method in `Schema.DescribeFieldResult` now indicates whether the field can be edited by the current user, or, if the field is a master-detail relationship field, whether the child records can be reparented to different parent records (`true`), or not (`false`).

DML Support for Permission Set sObjects

DML operations are now supported with the following sObjects:

- FieldPermissions
- ObjectPermissions
- PermissionSet
- PermissionSetAssignment
- SetupEntityAccess

With this added support, you can now perform DML operations such as create, update, and delete on these objects from Apex. This allows you to use Apex to manage permissions in your organization programmatically.

JSON Parsing Enhancements

JSON support in Apex allows you to parse and generate JSON-encoded content. In Summer '12, JSON methods for deserializing JSON content now perform more lenient parsing by default. They don't throw any errors if the JSON input string contains attributes, such as fields or objects, that don't exist in the type of the deserialized output object. They simply ignore these extra attributes when deserializing the JSON string. Note that the default behavior of these methods for Apex saved using Salesforce.com API version 24.0 or earlier hasn't changed. This means that the extra attributes aren't ignored and cause a run-time exception. The affected methods are:

- `deserialize` in `System.JSON`
- `readValueAs` in `System.JSONParser`

Also, new methods have been added that perform strict parsing when deserializing JSON content. Unlike their counterpart methods, these methods do throw a run-time exception if the JSON content contains extra attributes that aren't present in the deserialized type. These methods are:

- `deserializeStrict` in `System.JSON`
- `readValueAsStrict` in `System.JSONParser`

Using the new `deserializeUntyped` method of `System.JSON`, you can now deserialize any JSON-encoded content without having to specify a type. This allows you to deserialize JSON content regardless of the underlying types it contains. Also, previously, it wasn't possible to deserialize JSON content into an Object type or a Map containing Object values (for example, Map<String, Object>). This new method allows you to do so.

The following is a description of the `deserializeUntyped` method and an example.

Description	Deserializes the specified JSON string into collections of primitive data types.
Arguments	String *`jsonString`*

Return Type	Any type
Example	The following example deserializes a JSON representation of an appliance object into a map that contains primitive data types and further collections of primitive types. It then verifies the deserialized values.

```
String jsonInput = '{\n' +
    ' "description" :"An appliance",\n' +
    ' "accessories" : [ "powerCord", ' +
      '{ "right":"door handle1", ' +
        '"left":"door handle2" } ],\n' +
    ' "dimensions" : ' +
      '{ "height" : 5.5 , ' +
        '"width" : 3.0 , ' +
        '"depth" : 2.2 },\n' +
    ' "type" : null,\n' +
    ' "inventory" : 2000,\n' +
    ' "price" : 1023.45,\n' +
    ' "isShipped" : true,\n' +
    ' "modelNumber" : "123"\n' +
    '}';

Map<String, Object> m =
   (Map<String, Object>)
      JSON.deserializeUntyped(jsonInput);

System.assertEquals(
   'An appliance', m.get('description'));

List<Object> a =
   (List<Object>)m.get('accessories');
System.assertEquals('powerCord', a[0]);
Map<String, Object> a2 =
   (Map<String, Object>)a[1];
System.assertEquals(
   'door handle1', a2.get('right'));
System.assertEquals(
   'door handle2', a2.get('left'));

Map<String, Object> dim =
   (Map<String, Object>)m.get('dimensions');
System.assertEquals(
   5.5, dim.get('height'));
System.assertEquals(
   3.0, dim.get('width'));
System.assertEquals(
   2.2, dim.get('depth'));

System.assertEquals(null, m.get('type'));
System.assertEquals(
   2000, m.get('inventory'));
System.assertEquals(
   1023.45, m.get('price'));
System.assertEquals(
   true, m.get('isShipped'));
System.assertEquals(
   '123', m.get('modelNumber'));
```

Callout Limit Increase for Batch Apex

The limit of callouts (HTTP requests or Web services calls) for each method execution in a batch Apex class increased from 1 to 10. The `start`, `execute`, and `finish` methods can make up to 10 callouts each.

Apex Test Execution and Code Coverage Results Changes

You can now reset code coverage results for all classes and triggers in your organization to 0% from the Apex Classes page. To do so, click **Develop** > **Apex Classes** > **Clear Code Coverage**.

Also, we've added two new options to give you more control over how code coverage results are computed and how asynchronous Apex tests are executed from the Apex Test Execution page. To access these options, click **Develop** > **Apex Test Execution** > **Options...**.

- **Store Only Aggregated Code Coverage**: Stores code coverage results aggregated for all test methods. This option reduces code coverage calculation time when your organization contains large volumes of test methods and Apex code, that is, when the number of test methods multiplied by the number of all classes and triggers is in the amount of hundreds of thousands. With this option enabled, you can't view code coverage results for an individual test method. Also, if you modify test methods that are defined in a test class (a class annotated with `@isTest`), you'll have to clear code coverage results by clicking **Clear Code Coverage** on the Apex Classes page. This is because the existing code coverage results of classes and triggers that these test methods cover aren't automatically cleared when using this option and modifying the tests. This option applies to test methods regardless of how they're executed—synchronously or asynchronously.
- **Disable Parallel Apex Testing**: Executes asynchronous tests one at a time. This helps prevent test interference on shared data when tests run at the same time and access the same data. This only occurs when tests don't create their own data and turn off data isolation to access the organization's data. This option doesn't affect the asynchronous execution of tests, which continue to run asynchronously from the Apex Test Execution page.

New Type Methods

Two new instance methods have been added to `System.Type`. Using `getName`, you can now get the Type name that corresponds to a Type instance. Also, `newInstance` allows you to create an instance of the current Type. Using the new `newInstance` method with the existing `forName` method, package subscribers can implement interface methods of an interface provided in a package and have the package call their implementation. The following are the method descriptions and a sample.

Method	Return Type	Description

`getName`	`String`	Returns the name of the current type.
		
		This example shows how to get a Type's name. It first obtains a Type by calling `forName`, then calls `getName` on the Type object.
		
		`Type t =`
		`  Type.forName('MyClassName');`
		
		`String typeName =`
		`  t.getName();`
		`System.assertEquals('MyClassName',`
		`  typeName);`

`newInstance`	Any type	Creates an instance of the current type and returns this new instance. This method enables you to instantiate a Type that implements an interface and call its methods while letting someone else provide the methods' implementation. This example shows how to create an instance of a Type. It first gets a Type by calling `forName` with the name of a class, then calls `newInstance` on this Type object. The `newObj`

Method	Return Type	Description

		instance is declared with the interface type that the `ShapeImpl` class implements.
		`Type t = Type.forName('ShapeImpl');`
		`Shape newObj = t.newInstance();`

Sample: Instantiating a Type Based on Its Name

The following sample shows how to use the Type methods to instantiate a Type based on its name.

In this sample, `Vehicle` represents the interface that the `VehicleImpl` class implements. The last class contains the code sample that invokes the methods implemented in `VehicleImpl`.

This is the `Vehicle` interface.

```
global interface Vehicle {
    Long getMaxSpeed();
    String getType();
}
```

This is the implementation of the `Vehicle` interface.

```
global class VehicleImpl implements Vehicle {
    global Long getMaxSpeed() { return 100; }
    global String getType() { return 'Sedan'; }
}
```

The method in this class gets the name of the class that implements the `Vehicle` interface through a custom setting value. It then instantiates this class by getting the corresponding type and calling the `newInstance` method. Next, it invokes the methods implemented in `VehicleImpl`. This sample requires that you create a public list custom setting named `CustomImplementation` with a text field named `className`. Create one record for this custom setting with a data set name of `Vehicle` and a class name value of `VehicleImpl`.

```
public class CustomerImplInvocationClass {

    public static void invokeCustomImpl() {
        // Get the class name from a custom setting.
        // This class implements the Vehicle interface.
        CustomImplementation__c cs = CustomImplementation__c.getInstance('Vehicle');

        // Get the Type corresponding to the class name
        Type t = Type.forName(cs.className__c);

        // Instantiate the type.
        // The type of the instantiated object
        //   is the interface.
        Vehicle v = (Vehicle)t.newInstance();

        // Call the methods that have a custom implementation
        System.debug('Max speed: ' + v.getMaxSpeed());
        System.debug('Vehicle type: ' + v.getType());
    }
}
```

Change in Running Asynchronous Processes in Test Methods

Previously, batch and scheduled Apex, which run asynchronously, executed in test methods only when included within the `startTest` and `stopTest` Test methods. They executed after `stopTest`, and then the test could be verified. These asynchronous processes now execute at the end of test methods when you don't specify the `startTest` and `stopTest` methods. This change applies only to Apex that is saved using Salesforce.com API version 25.0 and later, but not in earlier versions. Note that future methods (methods annotated with `@future`) continue to execute at the end of test methods when they're not included within the `startTest` and `stopTest` methods.

Sorting Support for Non-Primitive Data Types in Lists

You can now sort Lists that contain non-primitive data types, such as sObjects or user-defined types, using the `List.sort` method. By default, you can sort Lists of sObjects. To sort Lists of your own Apex types, you must implement the new built-in `Comparable` interface with its `compareTo` method in your Apex class.

The `Comparable` interface contains the following method.

Name	Arguments	Return Type	Description
`compareTo`	Any type *`objectToCompareTo`*	Integer	Returns an Integer value that is the result of the comparison. The implementation of this method should return the following values: • 0 if this instance and *`objectToCompareTo`* are equal • > 0 if this instance is greater than *`objectToCompareTo`* • < 0 if this instance is less than *`objectToCompareTo`*

To implement the `Comparable` interface, you must first declare a global class with the `implements` keyword as follows:

```
global class Employee implements Comparable {
```

Next, your class must provide an implementation for the following method:

```
global Integer compareTo(Object compareTo) {
    // Your code here
}
```

This is an example implementation of the `Comparable` interface. The `compareTo` method in this example compares the employee of this class instance with the employee passed in the argument. The method returns an Integer value based on the comparison of the employee IDs.

```
global class Employee implements Comparable {

    public Long id;
    public String name;
    public String phone;

    // Constructor
    public Employee(Long i, String n, String p) {
        id = i;
        name = n;
        phone = p;
    }

    // Implement the compareTo() method
    global Integer compareTo(Object compareTo) {
        Employee compareToEmp = (Employee)compareTo;
```

```
        if (id == compareToEmp.id) return 0;
        else if (id > compareToEmp.id) return 1;
        else return -1;
    }
}
```

Active Query Cursor Limit Increase

The limit of active query cursors per user increased from 5 to 50. All Apex code has this higher limit except for the batch Apex `start` method, which has a limit of 5 open cursors per user. The remaining batch methods have the higher limit of 50 cursors.

With this higher limit, SOQL queries in your code are less likely to reach the cursor limit and cause errors resulting from having cursors released by the system. Query cursors are created when SOQL queries execute and make use of inner queries or generate large result sets.

Cursor limits for different Database.com features are tracked separately. For example, you can have 50 Apex query cursors and 50 batch cursors open at the same time.

Change in Sending Emails for Unhandled Exceptions

Starting in Summer '12, there has been a change concerning sending error emails to developers for unhandled exceptions. If the exception error is a duplicate, the corresponding email isn't suppressed and is still sent for Apex code running asynchronously—batch Apex, scheduled Apex, or future methods (methods annotated with `@future`.) Also, the subject of the error email now includes the job ID for asynchronous jobs. For Apex code running synchronously, duplicate error emails may get suppressed.

API Enhancements

Summer '12 (API version 25.0) improvements:

Changes across the API layer:

Changes to individual APIs:

New and Changed Objects

For information about new and changed Chatter objects, see Chatter API Objects on page 27.

New Objects

These objects are new in API version 25.0.

- The SetupEntityAccess object represents enabled Apex class access settings for the parent PermissionSet. This means you can use the API to query these settings in profiles and permission sets.

Changed Objects

These objects have changed in API version 25.0.

- You can now use the PermissionSet object to query users' permissions and access settings in their profile as well as assigned permission sets. The PermissionSet object now includes the following read-only fields, which associate every profile with a permission set that stores the profile's user, object, and field permissions, as well as setup entity access settings.
  - `IsOwnedByProfile`—specifies whether the permission set is owned by a profile (`true`) or not (`false`).
  - `ProfileId`—when the permission set is owned by a profile, specifies the ID of the Profile.

Chatter API Objects

Changed Chatter Objects and Chatter-Related Fields

These objects have changed in API version 25.0.

- On the ContentDocument object, the following fields have changed:
  - `OwnerId`—This field is now updateable.
  - `ParentId`—This field is now nillable.

- On the ContentDocumentLink object, the following fields have changed:
  - The `ContentDocumentId` and `LinkedEntityId` fields are now createable.
  - The `ShareType` field is now required.

  In addition to the field changes, the ContentDocumentLink object now supports the `update()` call and ContentDocumentLink objects with a `LinkedEntityId` of types `User`, `Group`, or `Organization` can be created and deleted via the API.
- On the User object, the following fields have changed:
  - `UserPreferencesDisableFileShareNotificationsForApi`—This field is now available for API version 25.0 and later.

    When `false`, email notifications are sent from the person who has shared a file to the users with whom the file has been shared.

Additional Enhancements

- You can now query and retrieve lead feed items that are associated with a converted lead record using the LeadFeed, NewsFeed, and UserProfileFeed 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 the *Database.com SOQL and SOSL Reference Guide*.

SOQL OFFSET—Generally Available

SOQL OFFSET enables you to specify the starting row of the result set your SOQL query returns. Using SOQL OFFSET is helpful for paging through large result sets when you need to quickly jump to a particular subset of the entire results. As the offset calculation is done on the server and only the result subset is returned, using `OFFSET` is more efficient than retrieving the full result set and then filtering the results locally. For example, this SOQL query returns a result set that skips the first 10 rows:

```
SELECT Name
FROM Merchandise__c
WHERE Price__c > 5.0
ORDER BY Name
LIMIT 100
OFFSET 10
```

SOAP API Enhancements

Changed Calls

These calls have changed in API version 25.0.

Call	Argument or Result Object	Field	Change	Description
`describeSObjects()`	`DescribeSObjectResult`	`updateable` property of the Field object obtained from the `DescribeSObjectResult`	Now also indicates whether child records in a master-detail relationship field can be reparented to different parent records.	Indicates whether the field can be edited by the current user, or, if the field is a master-detail relationship field, the child records can be reparented to different parent records (`true`), or not (`false`).

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.

Streaming API Enhancements

New Events Usage Field

A new field on the Company Information page lets you see at a glance:

- How many Streaming API events your organization has used within the last 24 hours (from the current time)
- The maximum number of events within a 24–hour period that your organization has

To see the `Streaming API Events` field, navigate to **Company Profile** > **Company Information**.

Metadata API Enhancements

Metadata REST API—Pilot

The Metadata REST API allows you to retrieve your organization's customization information, such as custom object definitions and page layouts, using the simplified approach of REST. In Summer '12, the service allows you to read metadata. It will be expanded in the future to write, update, and delete. The goal of this pilot is to allow developers to experience the new API and provide feedback that will guide implementation.

Note: Because this is a pilot, changes will be made to improve the service that may alter the current resources. Pilot features are for testing and learning purposes only. Please don't use this pilot API for production projects.

To enable the Metadata REST API for your organization, contact salesforce.com. We appreciate your interest and look forward to your feedback.

For more information, see the Metadata REST API Developer's Guide.

New Metadata Types

These metadata types are new in Metadata API version 25.0.

Metadata Type	Description
SecuritySettings	Represents an organization's security settings. Security settings define trusted IP ranges for network access, password and login requirements, and session expiration and security settings. **Note:** This metadata type is only supported by the `retrieve()`, `describeMetadata()`, and `listMetadata()` calls. It's not supported by the `deploy()`, `create()`, `delete()`, or `update()` calls. You can deploy this type with all the retrieved settings or with any retrieved settings removed except for `ipRanges`. However, you can't edit any settings. If you deploy with any changed values or any incorrect `ipRanges`, an error message is returned.

Updated Metadata

These metadata fields have been added or changed in Metadata API version 25.0.

Metadata Type or Related Object	Field	Change	Description
CustomApplication	`customApplicationComponents`	New	Represents custom console components (Visualforce pages) assigned to a Service Cloud console app.
CustomApplication	`detailPageRefreshMethod`	New	Determines how detail pages refresh in a Service Cloud console app.
CustomApplication	`domainWhitelist`	New	Represents any external domains that users can access from within a Service Cloud console app. For example, `www.yourdomain.com`.
CustomApplication	`isServiceCloudConsole`	New	Indicates if the application is a Service Cloud console app.
CustomApplication	`listPlacement`	New	Represents how lists display in a Service Cloud console app. Required if `isServiceCloudConsole` is `true`.
CustomApplication	`listRefreshMethod`	New	Determines how lists refresh in a Service Cloud console app.

Metadata Type or Related Object	Field	Change	Description
CustomApplication	`workspaceMappings`	New	Represents how records open in a Service Cloud console app. Required if `isServiceCloudConsole` is `true`.
CustomField	`reparentableMasterDetail`	New	Indicates whether the child records in the master-detail relationship can be reparented to different parent records. The default value is `false`.
CustomField	`deleteConstraint`	New	Provides deletion options for lookup relationships. Valid values are: **`SetNull`** This is the default. If the lookup record is deleted, the lookup field is cleared. **`Restrict`** Prevents the record from being deleted if it's in a lookup relationship. **`Cascade`** Deletes the lookup record as well as associated lookup fields.
DashboardComponent	`chartSummary`	New	Specifies the summary field for the chart data. Required if `isAutoSelectFromReport` is set to `false`.
DashboardComponent	`componentType`	Changed	This existing field has the following new valid values: • `ColumnLine` • `ColumnLineGrouped` • `ColumnLineStacked` • `ColumnLineStacked100`
DashboardComponent	`groupingColumn`	New	Specifies the field by which to group data. This data is displayed on the X-axis for vertical column charts and on the Y-axis for horizontal bar charts.
Profile	`loginHours`	New	Indicates the hours within which a user with this profile may log in. If not specified, the profile doesn't restrict a user's login hours.
Report (a new field on ReportAggregate)	`isCrossBlock`	New	Determines whether the custom summary formula is a cross-block formula, which is available with joined reports. `true` indicates a cross-block custom summary formula.

Metadata Type or Related Object	Field	Change	Description
			`false` indicates a standard custom summary formula.
Report (a new field on ReportBlockInfo)	`blockID`	New	Required. `blockId` is used in cross-block custom summary formulas and joined report charts to identify the block containing each summary field. `blockId` is assigned automatically. Valid values are B1 through B5.

Additional Database.com Enhancements

Enhanced Profile User Interface: Object and Tabs Page Renamed

In Summer ’12, the Objects and Tabs page in the enhanced profile user interface is now named “Object Settings.” This page has the same functionality—only the name has changed.

If the enhanced profile user interface is enabled in your organization, you can view this page by clicking **Manage Users** > **Profiles**, selecting a profile, then clicking **Object Settings**.

Login Access Policy

User Permissions Needed	
To set login access policies:	“Manage Users”
To log in as any user:	“Modify All Data”

With Summer ’12, administrators can log in as any user in their organization without asking end-users to grant login access.

To enable this login access policy:

1. Contact salesforce.com to have this feature enabled in your organization. You can’t complete the remaining steps until salesforce.com enables the feature.
2. Click **Security Controls** > **Login Access Policies**.
3. On the Login Access Policies page, enable **Administrators Can Log in as Any User**.
4. Click **Save**.

When this is enabled, users won’t have the option to grant login access to administrators, but they can still grant login access to salesforce.com or partner support agents, depending on the login access policies specified for the organization.

Regardless of the login access policy, when an administrator logs in as another user, the login and logout events are recorded in the setup audit trail.

Allow Reparenting Option in Master-Detail Relationship Definitions

In Summer ’12, administrators can now allow child records in master-detail relationships to be reparented to different parent records by selecting the `Allow reparenting` option in the master-detail relationship definition. By default, records in master-detail relationships can’t be reparented.

Object Limits Displayed on Custom Object Definitions

In Summer ’12, a new related list at the bottom of custom object definition pages displays information about the usage of:

- Custom fields
- Roll-up summary fields
- Custom relationship fields
- Formulas — unique relationships per object
- Active workflow rules
- Total workflow rules
- Approval processes
- Active lookup filters

- Active validation rules
- VLOOKUP functions
- Sharing rules (Both owner- and criteria-based)
- Sharing rules (Criteria-based only)

When an item exceeds the limit allowed for the object, or reaches 75% of the limit, a tip displays that suggests what you can do next.

Note: The object limit percentages displayed for items used are truncated, not rounded. For example, if your organization uses 95.55% of the limit for a particular item, the percentage displayed is 95%.

Changes in Staging Test Database Configuration

When you create or refresh a staging test database in Summer '12, you can configure it not to copy some data that is generally not useful in a test database. Keeping the minimum selections will speed up your test database copy.

- The setup audit trail history of your production organization won't be copied to your test database. The audit trail for your test database organization will start when you begin to use it.
- Password history (users' previous passwords) won't be copied to your test database.

Correction toTest Database Retention Policy

The Spring '12 documentation incorrectly said that before an unactivated or locked test database is scheduled for deletion, you'll receive at least three email notifications. You'll get two emails reminding you to activate or obtain additional licenses before these test databases are deleted.

New Lookup Relationship Options

Available in: **All** Editions.

Starting in Summer '12, you have new choices when you create a lookup relationship between two records, giving you more control over how you define the relationship. First, you can make the lookup field required or optional. If the lookup field is optional, you can specify one of three behaviors to occur if the lookup record is deleted:

- `Clear the value of this field` This is the default. Clearing the field is a good choice when the field does not have to contain a value from the associated lookup record.
- `Don't allow deletion of the lookup record that's part of a lookup relationship` This option restricts the lookup record from being deleted if you have any dependencies, such as a workflow rule, built on the relationship.
- `Delete this record also` Available only if a custom object contains the lookup relationship, not if it's contained by a standard object. However, the lookup object can be either standard or custom. Choose when the lookup field and its associated record are tightly coupled and you want to completely delete related data.

Caution: Choosing `Delete this record also` can result in a *cascade-delete*. A cascade-delete bypasses security and sharing settings, which means users can delete records when the target lookup record is deleted *even if they don't have access to the records*. To prevent records from being accidentally deleted, contact Salesforce.com to get the cascade-delete option enabled for your organization.

Formula Function Changes

We have improved how we compile many formula functions. Compiled formula fields still cannot exceed 5,000 bytes, but individual functions (such as DATEVALUE) now contribute less to the function's total compile size. This should improve the performance of affected formulas.

We also made some improvements around error checking for the DATE function.

Finally, we've improved the documentation for several formula functions on the Operators and Functions online help:

- CASESAFEID has been added.
- The Description is improved for the following functions:
  - BR
  - IMAGE
  - ISNEW
  - JSENCODE
  - JSINHTMLENCODE
  - REGEX
  - URLENCODE

Supported Locale Online Help Enhanced

We've added three locales to the Supported Locales documentation: Hungarian (Hungary), Indonesian (Indonesia), and Turkish (Turkey).

Custom field labels changes

We now support only plain text in custom field labels for organizations not currently using HTML in their labels.

Time-Dependent Field Updates Can Retrigger Workflow Rules

In Summer '12, field update actions for time-based workflow rules can trigger the re-evaluation of workflow rules. Previously, time-dependent field updates that caused field values to change didn't trigger workflow rule re-evaluation on the associated objects.