

salesforce



Salesforce.com: Spring '12

Database.com User Guide



Last updated: January 19 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

Database.com User Guide.....	2
Managing Your Organization.....	2
Personalizing Database.com.....	73
Managing the Database Schema.....	85
Managing Users.....	194
Managing Security.....	235
Importing and Exporting Data.....	346
Managing Application Logic.....	379
Developing with APIs.....	425
Developing and Testing in a Test Database Organization.....	429
Working with the Developer Console.....	445
Index.....	463

Database.com User Guide

Managing Your Organization

Setting Up Your Organization

About Company Information

User Permissions Needed	
To view company information:	“View Setup and Configuration”
To change company information:	“Customize Application”

When your company signs up, the information provided during signup is displayed on the Company Information page at **Company Profile > Company Information**.

From the Company Information page, you can:

- Click **Edit** to change your company's information, including your organization's Default Language setting.
- Click **Currency Setup** to set up the ability to use multiple currencies.
- Go to Checkout to buy additional user or feature licenses by clicking **Buy More Licenses** in the appropriate related list. For detailed instructions, see [Checkout User Guide](#).

The Company Information page also displays all of the active user and feature licenses you have purchased for your organization. A user license entitles a user to different functionality within Database.com and determines which profiles and permission sets are available to the user.

This page lists the following for each type of license:

- Status indicates the status of the license.
- Total Licenses indicates the number of licenses for which your company is billed and that are available to you.
- Used Licenses is the number of licenses that you have assigned to users.
- Remaining Licenses is the number of unused licenses.

Setting Up Your Organization

Click **Administration Setup** to open the Administration Setup page. It contains setup and customization options to help you set up your Database.com organization.

The sidebar includes tools for [browsing and searching setup options](#).

Manage Users

Expand the **Manage Users** folder under **Administration Setup** to access the following options:

Users

Create, edit, and deactivate users.

Roles

Define your organization's role hierarchy.

Permission Sets

Specify user permissions, object permissions, field permissions, and access to Apex classes and service providers, without changing users' profiles.

Profiles

Specify user permissions, object permissions, field permissions, login settings, and access to Apex classes and service providers.

Public Groups

Add, update, or delete public groups.

Queues

Add, update, or delete queues for custom objects.

Login History

View when your users are logging in.

Training History

View which users have taken online training.

Company Profile

Expand the **Company Profile** folder under **Administration Setup** to access the following options:

Company Information

Update your company's information and set up multiple currencies.

Manage Currencies

Set up multiple currencies.

My Domain

Set up a custom Database.com domain name, which appears in the URLs that you use to login to and use the application.

Security Controls

Expand the **Security Controls** folder under **Administration Setup** to access the following options:

Sharing Settings

Define how your users share data.

Field Accessibility

View the access that users have to specific fields based on profile.

Password Policies

Define password policies for security.

Session Settings

Lock users' sessions to an IP address or change session timeout settings.

Network Access

Define IP addresses from which users can log in to your organization.

Certificate and Key Management

Create and manage Database.com key pairs and certificates for your organization.

View Setup Audit Trail

View which users have recently changed your organization's setup.

Expire All Passwords

Expire passwords for all users in the organization.

Remote Site Settings

Specify the Web addresses that your organization can invoke from Database.com. You must specify a site before callouts to them from Apex or the AJAX proxy will function correctly.

Data Management

Expand the **Data Management** folder under **Administration Setup** to access the following options:

Storage Usage

View how much data storage and file storage your organization is using.

Mass Transfer Records

Transfer multiple records at one time.

Test Database

Create a complete single copy of your organization in a separate environment to do a variety of actions—such as quality assurance testing, integration testing, or user training—withoucompromising your organization's data.

Data Loader

Download a client application that allows you to import, update, delete, and export large quantities of records.

Monitoring

Expand the **Monitoring** folder under **Administration Setup** to access the following options:

Outbound Messages

An administrator can view the Outbound Message queue to check the status of outbound messages related to workflow.

Time-Based Workflow

Specify criteria for monitoring the workflow queue, which contains pending actions triggered by workflow rules.

API Usage Notifications

Define a notification process that automatically sends email to a specified user when API requests for an organization exceed the specified limit.

Debug Logs

Specify the users for whom you want to retain the Apex debug logs in your organization.

Scheduled Jobs

View all the jobs scheduled to run by users.

Bulk Data Load Jobs

Monitor the status of current and recent bulk data load jobs.

Company Information Fields

The Company Information page has the following fields (listed in alphabetical order), including the [user](#) and [feature](#) licenses purchased for your organization.

Field	Description
Address	Street address of the company. Up to 255 characters are allowed in this field.
Admin Newsletter	Allow administrators in your organization to choose whether they want to receive administrator-targeted promotional emails from salesforce.com.
API Requests, Last 24 Hours	The total number of API requests issued by the organization in the last 24 hours. For more information about API usage notification, see About API Usage Notifications on page 427 or the Web Services API Developer's Guide .
Corporate Currency	The base rate from which the application's other currencies are converted. Serves as the basis for all currency conversion rates. Only for organizations that use multiple currencies.
Created By	User who signed up the organization, including creation date and time. (Read only)
Default Language	
Default Locale	The default country or geographic region that is selected for new users in the organization.
Default Time Zone	Primary time zone in which the organization is located. A user's individual Time Zone setting overrides the organization's Default Time Zone setting. Note: Organizations in Arizona should select "Mountain Standard Time," and organizations in parts of Indiana that do not follow Daylight Savings Time should select "Eastern Standard Time."
Fax	Fax number. Up to 40 characters are allowed in this field.
Hide Notices About System Downtime	Select this checkbox to prevent advance notices about planned system downtime from displaying to users when they log in to Database.com.
Hide Notices About System Maintenance	Select this checkbox to prevent advance notices about planned system maintenance from displaying to users when they log in to Database.com.
Modified By	User who last changed the company information, including modification date and time. (Read only)

Field	Description
Newsletter	Allow users in your organization to choose whether they want to receive user-targeted promotional emails from salesforce.com.
Organization Name	Name of the organization. Up to 80 characters are allowed in this field.
Phone	Main phone number at organization. Up to 40 characters are allowed in this field.
Primary Contact	Person who is main contact or administrator at the organization. You can enter a name, or select a name from a list of previously defined users. Up to 80 characters are allowed in this field.
Salesforce.com Organization ID	Code that uniquely identifies your organization to salesforce.com.
Restricted Logins, Current Month	Number of restricted login users who have logged in during the current month. This value resets to zero at the beginning of each month. The maximum number of restricted login users for the organization is in parentheses.
Used Records	Number of records in use and percentage of the total number of records available (for example, 10%).
Used File Space	Amount of file storage in use; the value is expressed as a measurement (for example, 500 MB) and as a percentage of the total amount of file storage available (for example, 10%).

Managing Currencies, Time Zones, and Locales

Understanding Language, Locale, and Currency

User Permissions Needed	
To view company information:	“View Setup and Configuration”
To change company information	“Customize Application”

The settings for language, locale, time zone, and currency affects how data is handled. In a single currency organization, the Database.com administrators set the currency locale, default language, default locale, and default time zone for their organizations and the language, locale, and time zone can be set for the user. In a multiple currency organization, the Database.com administrators set the corporate currency, default language, default locale, and default time zone for their organizations and the users can set their individual currency, language, locale, and time zone.



Note: Single language organizations cannot change their language, although they can change their locale.

Setting	Who can edit the setting	How to edit the setting	Description and effects of the setting
Currency	User in a multiple currency organization	Go to My Personal Information > Personal Information > Edit and select a currency from the activated currencies drop-down list.	User's default currency. Shown only in organizations using multiple currencies. This must be one of the active currencies for the organization.
Corporate Currency	Administrator in a multiple currency organization	Go to Company Profile > Manage Currencies > Change Corporate and select a currency from the New Corporate Currency drop-down list. To add a currency, click New , select a currency from the supported currency drop-down list, enter a conversion rate, enter the number of decimal places you want, and click Save or Save & New to add another currency.	The base rate from which the application's other currencies are converted. Serves as the basis for all currency conversion rates. Only for organizations that use multiple currencies.
Default Currency ISO Code	Not editable	Not editable	User's default currency setting for new records. Available only for organizations that use multiple currencies.
Default Language	Administrator	Go to Company Profile > Company Information > Edit and select a language from the supported language drop-down list.	For Database.com Admin users, the default language that is selected for new users of the Database.com Console. For users with the Database.com User and Database.com Light User license types, this setting has no effect. You can, however, use values that users with the Database.com User and Database.com Light User license types entered into the Default Language field to create a translation infrastructure for your application.

Setting	Who can edit the setting	How to edit the setting	Description and effects of the setting
Default Locale	Administrator	<p>Go to Company Profile > Company Information > Edit and select a locale from the supported locale drop-down list.</p>  <p>Note: Locale names with a country in parentheses also set a default currency</p>	<p>The default country or geographic region that is selected for new users in the organization.</p>
Default Time Zone	Administrator	<p>Go to Company Profile > Company Information > Edit and select a time zone from the supported time zone drop-down list.</p>	<p>Primary time zone in which the organization is located. A user's individual Time Zone setting overrides the organization's Default Time Zone setting.</p> <p>Note: Organizations in Arizona should select "Mountain Standard Time," and organizations in parts of Indiana that do not follow Daylight Savings Time should select "Eastern Standard Time."</p>
Language	User	<p>Go to My Personal Information > Personal Information > Edit and select a language from the supported language drop-down list.</p>	
Locale	User	<p>Go to My Personal Information > Personal Information > Edit and select a locale from the supported locale drop-down list.</p>  <p>Note: Locale names with a country in parentheses also set a default currency</p>	<p>Country or geographic region in which user is located.</p> <p>The Locale setting affects the format of date, date/time, and number fields. For example, dates in the English (United States) locale display as 06/30/2000 and as 30/06/2000 in the English (United Kingdom) locale. Times in the English (United States) locale display using a twelve-hour clock with AM and PM (for example, 2:00 PM), whereas in the English (United Kingdom) locale, they display using a</p>

Setting	Who can edit the setting	How to edit the setting	Description and effects of the setting
Time Zone	User	Go to My Personal Information > Personal Information > Edit and select a time zone from the supported time zone drop-down list.	Twenty-four-hour clock (for example, 14:00). The <code>Locale</code> setting also affects the first and last name order on Name fields for users. For example, Bob Johnson in the English (United States) locale displays as Bob Johnson, whereas the Chinese (China) locale displays the name as Johnson Bob.
			Primary time zone in which user works. Users in Arizona should select the setting with “America/Phoenix,” and users in parts of Indiana that do not follow Daylight Savings Time should select the setting with “America/Indianapolis.”

Managing Multiple Currencies

User Permissions Needed	
To view currencies:	“View Setup and Configuration”
To change currencies:	“Customize Application”

You can set up your organization to store multiple currencies. To do so:

1. Contact salesforce.com to enable Multi-Currency for your organization.



Note: For large organizations, Multi-Currency might be enabled over the next weekend to avoid performance issues during the work week.

2. Designate your corporate currency. See [Setting Corporate Currency](#) on page 11.
3. Activate additional currencies for your organization. See [Activating and Deactivating Currencies](#) on page 10.
4. Set conversion rates for new currencies. See [Editing Conversion Rates](#).

Using Multiple Currencies

Custom formula fields are not tied to any particular currency. If the result of a formula field is a currency amount, it displays in the currency of the associated record. This is also true for cross-object formulas that reference merge fields with different

currencies and formulas in workflow rules. However, note that workflow rules that use filters instead of formulas convert all currency values to the corporate currency.

You cannot disable multiple currencies for your organization if the currency fields are referenced in an Apex script. For example, if a script references the `Salary_Currency` field (represented as `Position__c.MinPay__c` in the code), multiple currencies cannot be disabled. For more information, see [Apex Code Overview](#) on page 379.

Activating and Deactivating Currencies

User Permissions Needed	
To view currencies:	“View Setup and Configuration”
To change currencies:	“Customize Application”

To use multiple currencies, your administrator must specify which currencies are supported.

- **Active currencies**—These represent currencies that must be stored in your organization. Once you activate a currency, you can never permanently delete it.
- **Inactive currencies**—These are currencies that your organization no longer uses. You may have existing records that use inactive currencies, but you cannot enter new amounts in inactive currencies.

To activate new currencies:

1. Click **Company Profile > Manage Currencies**.
2. Click **New** in the Active Currencies related list.
3. Select a currency. Currencies are alphabetized using their ISO currency code.
4. Enter the conversion rate relative to your corporate currency.
5. Specify the number of decimal places to show for amounts in this currency.
6. Click **Save**.

To activate a currency from the list of inactive currencies, click **Activate** next to the currency.

To deactivate a currency, click **Deactivate** next to the currency. Deactivating a currency does not alter amounts in items that use that currency. However, you can no longer enter new amounts using the inactive currency.

Editing Conversion Rates

User Permissions Needed	
To view currencies:	“View Setup and Configuration”
To change currencies:	“Customize Application”

You can manage exchange rates between your active and inactive currencies and the corporate currency by editing the conversion rates. These are static exchange rates that apply to all currency fields used in your organization.

To edit your organization's static conversion rates:

1. Click **Company Profile > Manage Currencies**.
2. Click **Edit Rates** in the Active Currencies or Inactive Currencies lists.
3. Enter the conversion rate between each currency and your corporate currency.

4. Click **Save**.

When you change the conversion rates, currency amounts are updated using the new rates. Previous conversion rates are not stored.



Note:

- You cannot track revenue gain or loss based on currency fluctuations.
- Changing conversion rates causes a mass recalculation of **roll-up summary fields**, which may take up to 30 minutes, depending on the number of records affected and other factors.

Setting Corporate Currency

User Permissions Needed	
To view currencies:	“View Setup and Configuration”
To change currencies:	“Customize Application”

Your administrator must specify a Corporate Currency, which reflects the currency in which your corporate headquarters reports revenue. A salesforce.com representative initially sets your corporate currency upon activation of the feature.

To change your corporate currency:

1. Click **Company Profile > Manage Currencies**.
2. Click **Change Corporate**.
3. Choose a new corporate currency from the list of active currencies, and click **Save**. If you have not yet set up any currencies, see [Activating and Deactivating Currencies](#) on page 10.

The corporate currency is the currency on which all of your conversion rates are based.

Supported Currencies

User Permissions Needed	
To view company information:	“View Setup and Configuration”
To change company information:	“Customize Application”

Database.com supported currencies:

Currency Name	Currency Code
Afghanistan Afghani	AFN
Albanian Lek	ALL
Algerian Dinar	DZD
Angola Kwanza	AOA
Argentine Peso	ARS

Currency Name	Currency Code
Armenian Dram	AMD
Aruba Florin	AWG
Australian Dollar	AUD
Azerbaijanian Manat	AZN
Bahamian Dollar	BSD
Bahraini Dinar	BHD
Bangladesh Taka	BDT
Barbados Dollar	BBD
Belarussian Ruble	BYR
Belize Dollar	BZD
Bermuda Dollar	BMD
Bhutan Ngultrum	BTN
Bolivia Mvdol	BOV
Bolivian Boliviano	BOB
Botswana Pula	BWP
Brazilian Real	BRL
British Pound	GBP
Brunei Dollar	BND
Bulgaria Lev	BGN
Burundi Franc	BIF
Cambodia Riel	KHR
Canadian Dollar	CAD
Cape Verde Escudo	CVE
Cayman Islands Dollar	KYD
CFA Franc (BCEAO)	XOF
CFA Franc (BEAC)	XAF
Chilean Peso	CLP
Chinese Renminbi	CN
Chinese Yuan	CNY
Colombian Peso	COP
Comoros Franc	KMF

Currency Name	Currency Code
Convertible Marks	BAM
Costa Rica Colon	CRC
Croatian Kuna	HRK
Cuban Peso	CUP
Czech Koruna	CZK
Danish Krone	DKK
Djibouti Franc	DJF
Dominican Peso	DOP
East Caribbean Dollar	XCD
Egyptian Pound	EGP
Eritrea Nakfa	ERN
Estonian Kroon	EEK
Ethiopian Birr	ETB
Euro	EUR
Falkland Islands Pound	FKP
Fiji Dollar	FJD
Franc Congolais	CDF
Gambian Dalasi	GMD
Georgia Lari	GEL
Ghanian Cedi	GHS
Gibraltar Pound	GIP
Guatemala Quetzal	GTQ
Guinea Franc	GNF
Guyana Dollar	GYD
Haiti Gourde	HTG
Honduras Lempira	HNL
Hong Kong Dollar	HKD
Hungarian Forint	HUF
Iceland Krona	ISK
Indian Rupee	INR
Indonesian Rupiah	IDR

Currency Name	Currency Code
Iranian Rial	IRR
Iraqi Dinar	IQD
Israeli Shekel	ILS
Jamaican Dollar	JMD
Japanese Yen	JPY
Jordanian Dinar	JOD
Kazakhstan Tenge	KZT
Kenyan Shilling	KES
Korean Won	KRW
Kuwaiti Dinar	KWD
Kyrgyzstan Som	KGS
Lao Kip	LAK
Latvian Lat	LVL
Lebanese Pound	LBP
Lesotho Loti	LSL
Liberian Dollar	LRD
Libyan Dinar	LYD
Lithuanian Lita	LTL
Macau Pataca	MOP
Macedonian Denar	MKD
Malagasy Ariary	MGA
Malawi Kwacha	MWK
Malaysian Ringgit	MYR
Maldives Rufiyaa	MVR
Mauritania Ougulya	MRO
Mauritius Rupee	MUR
Mexican Peso	MXN
Mexican Unidad de Inversion (UDI)	MXV
Moldovan Leu	MDL
Mongolian Tugrik	MNT
Moroccan Dirham	MAD

Currency Name	Currency Code
Mozambique Metical	MZN
Myanmar Kyat	MMK
Namibian Dollar	NAD
Nepalese Rupee	NPR
Neth Antilles Guilder	ANG
New Zealand Dollar	NZD
Nicaragua Cordoba	NIO
Nigerian Naira	NGN
North Korean Won	KPW
Norwegian Krone	NOK
Omani Rial	OMR
Pacific Franc	XPF
Pakistani Rupee	PKR
Panama Balboa	PAB
Papua New Guinea Kina	PGK
Paraguayan Guarani	PYG
Peruvian Nuevo Sol	PEN
Philippine Peso	PHP
Polish Zloty	PLN
Qatar Rial	QAR
Romanian Leu	RON
Russian Rouble	RUB
Rwanda Franc	RWF
Samoa Tala	WST
Sao Tome Dobra	STD
Saudi Arabian Riyal	SAR
Serbian Dinar	RSD
Seychelles Rupee	SCR
Sierra Leone Leone	SLL
Singapore Dollar	SGD
Solomon Islands Dollar	SBD

Currency Name	Currency Code
Somali Shilling	SOS
South African Rand	ZAR
Sri Lanka Rupee	LKR
St Helena Pound	SHP
Sudanese Pound	SDG
Surinam Dollar	SRD
Swaziland Lilageni	SZL
Swedish Krona	SEK
Swiss Franc	CHF
Syrian Pound	SYP
Taiwan Dollar	TWD
Tajik Ruble	TJS
Tanzanian Shilling	TZS
Thai Baht	THB
Tonga Pa'anga	TOP
Trinidad&Tobago Dollar	TTD
Tunisian Dinar	TND
Turkish Lira	TRY
Turkmenistan Manat	TMT
U.S. Dollar	USD
UAE Dirham	AED
Ugandan Shilling	UGX
Ukraine Hryvnia	UAH
Unidades de fomento	CLF
Uruguayan New Peso	UYU
Uzbekistan Sum	UZS
Vanuatu Vatu	VUV
Venezuelan Bolivar Fuerte	VEF
Vietnam Dong	VND
Yemen Riyal	YER
Zambian Kwacha	ZMK

Currency Name	Currency Code
Zimbabwe Dollar	ZWD

Supported Time Zones

User Permissions Needed	
To view company information:	“View Setup and Configuration”
To change company information:	“Customize Application”

Database.com supported times zones and codes (in chronological order):

Time Zone Code	Time Zone Name
GMT+14:00	Line Is. Time (Pacific/Kiritimati)
GMT+13:00	Phoenix Is. Time (Pacific/Enderbury)
GMT+13:00	Tonga Time (Pacific/Tongatapu)
GMT+12:45	Chatham Daylight Time (Pacific/Chatham)
GMT+12:00	Petropavlovsk-Kamchatski Time (Asia/Kamchatka)
GMT+12:00	New Zealand Daylight Time (Pacific/Auckland)
GMT+12:00	Fiji Time (Pacific/Fiji)
GMT+11:30	Norfolk Time (Pacific/Norfolk)
GMT+11:00	Solomon Is. Time (Pacific/Guadalcanal)
GMT+10:30	Lord Howe Summer Time (Australia/Lord_Howe)
GMT+10:00	Eastern Standard Time (Queensland)
GMT+10:00	Eastern Summer Time (New South Wales)
GMT+09:30	Central Summer Time (South Australia)
GMT+09:30	Central Standard Time (Northern Territory)
GMT+09:00	Korea Standard Time (Asia/Seoul)
GMT+09:00	Japan Standard Time (Asia/Tokyo)
GMT+08:00	Hong Kong Time (Asia/Hong_Kong)
GMT+08:00	Malaysia Time (Asia/Kuala_Lumpur)
GMT+08:00	Philippines Time (Asia/Manila)
GMT+08:00	China Standard Time (Asia/Shanghai)
GMT+08:00	Singapore Time (Asia/Singapore)

Time Zone Code	Time Zone Name
GMT+08:00	China Standard Time (Asia/Taipei)
GMT+08:00	Western Standard Time (Australia)
GMT+07:00	Indochina Time (Asia/Bangkok)
GMT+07:00	West Indonesia Time (Asia/Jakarta)
GMT+07:00	Indochina Time (Asia/Saigon)
GMT+06:30	Myanmar Time (Asia/Rangoon)
GMT+06:00	Bangladesh Time (Asia/Dacca)
GMT+05:45	Nepal Time (Asia/Katmandu)
GMT+05:30	India Standard Time (Asia/Calcutta)
GMT+05:30	India Standard Time (Asia/Colombo)
GMT+05:00	Pakistan Summer Time (Asia/Karachi)
GMT+05:00	Uzbekistan Time (Asia/Tashkent)
GMT+05:00	Yekaterinburg Time (Asia/Yekaterinburg)
GMT+04:30	Afghanistan Time (Asia/Kabul)
GMT+04:00	Gulf Standard Time (Asia/Dubai)
GMT+04:00	Georgia Time (Asia/Tbilisi)
GMT+03:30	Iran Standard Time (Asia/Tehran)
GMT+03:00	Eastern African Time (Africa/Nairobi)
GMT+03:00	Arabia Standard Time (Asia/Baghdad)
GMT+03:00	Arabia Standard Time (Asia/Kuwait)
GMT+03:00	Arabia Standard Time (Asia/Riyadh)
GMT+03:00	Moscow Standard Time (Europe/Moscow)
GMT+02:00	Eastern European Time (Africa/Cairo)
GMT+02:00	South Africa Standard Time (Africa/Johannesburg)
GMT+02:00	Israel Standard Time (Asia/Jerusalem)
GMT+02:00	Eastern European Time (Europe/Athens)
GMT+02:00	Eastern European Time (Europe/Bucharest)
GMT+02:00	Eastern European Time (Europe/Helsinki)
GMT+02:00	Eastern European Time (Europe/Istanbul)
GMT+02:00	Eastern European Time (Europe/Minsk)
GMT+01:00	Central European Time (Europe/Amsterdam)

Time Zone Code	Time Zone Name
GMT+01:00	Central European Time (Europe/Berlin)
GMT+01:00	Central European Time (Europe/Brussels)
GMT+01:00	Central European Time (Europe/Paris)
GMT+01:00	Central European Time (Europe/Prague)
GMT+01:00	Central European Time (Europe/Rome)
GMT+00:00	Irish Summer Time (Europe/Dublin)
GMT+00:00	Western European Summer Time (Europe/Lisbon)
GMT+00:00	British Summer Time (Europe/London)
GMT+00:00	Greenwich Mean Time (GMT)
GMT-01:00	Cape Verde Time (Atlantic/Cape_Verde)
GMT-02:00	South Georgia Standard Time (Atlantic/South_Georgia)
GMT-03:00	Argentine Summer Time (America/Buenos_Aires)
GMT-03:00	Brasilia Summer Time (America/Sao_Paulo)
GMT-03:30	Newfoundland Daylight Time (America/St_Johns)
GMT-04:00	Atlantic Daylight Time (America/Halifax)
GMT-04:00	Atlantic Standard Time (America/Puerto_Rico)
GMT-04:00	Chile Summer Time (America/Santiago)
GMT-04:00	Atlantic Daylight Time (Atlantic/Bermuda)
GMT-04:30	Venezuela Time (America/Caracas)
GMT-05:00	Colombia Time (America/Bogota)
GMT-05:00	Eastern Daylight Time (America/Indianapolis)
GMT-05:00	Peru Time (America/Lima)
GMT-05:00	Eastern Daylight Time (America/New_York)
GMT-05:00	Eastern Standard Time (America/Panama)
GMT-06:00	Central Daylight Time (America/Chicago)
GMT-06:00	Central Standard Time (America/El_Salvador)
GMT-06:00	Central Standard Time (America/Mexico_City)
GMT-07:00	Mountain Daylight Time (America/Denver)
GMT-07:00	Mountain Standard Time (America/Phoenix)
GMT-08:00	Pacific Daylight Time (America/Los_Angeles)
GMT-08:00	Pacific Standard Time (America/Tijuana)

Time Zone Code	Time Zone Name
GMT-09:00	Alaska Daylight Time (America/Anchorage)
GMT-10:00	Hawaii Standard Time (Pacific/Honolulu)
GMT-11:00	Niue Time (Pacific/Niue)
GMT-11:00	Samoa Standard Time (Pacific/Pago_Pago)

Supported Locales

User Permissions Needed	
To view company information:	“View Setup and Configuration”
To change company information:	“Customize Application”

The Database.com locale settings determine the following display formats:

- Date and time
- Users' names
- Addresses
- Commas and periods in numbers

Locale names with a country in parentheses also set a default currency.

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
Arabic	ar		02/01/2008 04:30 PM	06:00	1,234.56	Ms. FName LName	Address Line 1, Address Line 2, City, State ZipCode Country
Arabic (United Arab Emirates)	ar_AE	UAE Dirham: AED	02/01/2008 04:30 PM	06:00	1,234.56	Ms. FName LName	Address Line 1, Address Line 2, City, State ZipCode Country

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
Arabic (Bahrain)	ar_BH	Bahraini Dinar: BHD	02/01/2008 04:30 PM	06:00	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Arabic (Egypt)	ar_EG	Egyptian Pound EGP	02/01/2008 04:30 PM	06:00	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Arabic (Jordan)	ar_JO	Jordanian Dinar: JOD	02/01/2008 04:30 PM	06:00	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Arabic (Kuwait)	ar_KW	Kuwaiti Dinar: KWD	02/01/2008 04:30 PM	06:00	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Arabic (Lebanon)	ar_LB	Lebanese Pound: LBP	02/01/2008 04:30 PM	06:00	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
Arabic (Saudi Arabia)	ar_SA	Saudi Arabian Riyal: SAR	02/01/2008 04:30 PM	06:00	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Bulgarian	bg		2008-1-2 16:30	6:00	1 234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Bulgarian (Bulgaria)	bg_BG	Bulgaria Lev: BGN	2008-1-2 16:30	6:00	1 234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Catalan	ca		02/01/2008 16:30	06:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Catalan (Spain)	ca_ES	Euro: EUR	02/01/2008 16:30	06:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
Czech	cs		2.1.2008 16:30	6:00	1 234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Czech (Czech Republic)	cs_CZ	Czech Koruna: CZK	2.1.2008 16:30	6:00	1 234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Danish	da		02-01-2008 16:30	06:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Danish (Denmark)	da_DK	Danish Krone: DKK	02-01-2008 16:30	06:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
German	de		02.01.2008 16:30	06:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 ZipCode City State Country

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
German (Austria)	de_AT	Euro: EUR	02.01.2008 16:30	06:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 ZipCode City State Country
German (Switzerland)	de_CH	Swiss Franc: CHF	02.01.2008 16:30	06:00	1'234.56	Ms. FName LName	Address Line 1, Address Line 2 ZipCode City State Country
German (Germany)	de_DE	Euro: EUR	02.01.2008 16:30	06:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 ZipCode City State Country
German (Luxembourg)	de_LU	Euro: EUR	02.01.2008 16:30	06:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 ZipCode City State Country
Greek	el		2/1/2008 4:30 PM	6:00 μ	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Greek (Greece)	el_GR	Greek Drachma: EUR	2/1/2008 4:30 PM	6:00 μ	1.234,56	Ms. FName LName	Address Line 1, Address Line 2

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
							City, State ZipCode Country
English (Australia)	en_AU	Australian Dollar: AUD	2/01/2008 4:30 PM	6:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
English (Barbados)	en_BB	Barbados Dollar: BBD	1/2/2008 4:30 PM	6:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
English (Bermuda)	en_BM	Bermuda Dollar: BMD	1/2/2008 4:30 PM	6:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
English (Canada)	en_CA	Canadian Dollar: CAD	02/01/2008 4:30 PM	6:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
English (United Kingdom)	en_GB	British Pound: GBP	02/01/2008 16:30	06:00	1,234.56	Ms. FName LName	Address Line 1, Address Line 2

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
							City, State ZipCode Country
English (Ghana)	en_GH	Ghanian Cedi (New): GHS	1/2/2008 4:30 PM	6:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
English (Indonesia)	en_ID	Indonesian Rupiah: IDR	1/2/2008 4:30 PM	6:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
English (Ireland)	en_IE	Euro: EUR	02/01/2008 16:30	06:00	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
English (India)	en_IN	Indian rupee: INR	2/1/2008 4:30 PM	6:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
English (Malaysia)	en_MY	Malaysian Dollar (Ringgit): MYR	1/2/2008 4:30 PM	6:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
							City, State ZipCode Country
English (Nigeria)	en_NG	Nigerian Naira: NGN	1/2/2008 4:30 PM	6:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
English (New Zealand)	en_NZ	New Zealand Dollar: NZD	2/01/2008 4:30 PM	6:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
English (Philippines)	en_PH	Philippines Peso: PHP	1/2/2008 4:30 PM	6:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
English (Singapore)	en_SG	Singapore Dollar: SGD	1/2/2008 4:30 PM	6:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
English (United States)	en_US	U.S. Dollar: USD	1/2/2008 4:30 PM	6:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
							City, State ZipCode Country
English (South Africa)	en_ZA	South African Rand: ZAR	2008/01/02 4:30 PM	6:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Spanish (Argentina)	es_AR	Argentine Peso: ARS	02/01/2008 16:30	6:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Spanish (Bolivia)	es_BO	Bolivian Boliviano: BOB	02-01-2008 04:30 PM	06:00 AM	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Spanish (Chile)	es_CL	Chilean Peso: CLP	02-01-2008 04:30 PM	06:00 AM	1.234,56	Ms. FName LName	Address Line 1, Address Line 2

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
							City, State ZipCode Country
Spanish (Colombia)	es_CO	Colombian Peso: COP	2/01/2008 04:30 PM	06:00 AM	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Spanish (Costa Rica)	es_CR	Costa Rica Colon: CRC	02/01/2008 04:30 PM	06:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Spanish (Dominican Republic)	es_DO	Dominican Republic Peso: DOP	01/02/2008 04:30 PM	06:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Spanish (Ecuador)	es_EC	CFA Franc (BEAC): XAF	02/01/2008 04:30 PM	06:00 AM	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Spanish (Spain)	es_ES	Euro: EUR	2/01/2008 16:30	6:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
							City, State ZipCode Country
Spanish (Guatemala)	es_GT	Guatemala Quetzal: GTQ	2/01/2008 04:30 PM	06:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Spanish (Honduras)	es_HN	Honduras Lempira: HNL	01-02-2008 04:30 PM	06:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Spanish (Mexico)	es_MX	Mexican Unidad de Inversion (UDI): MXV	2/01/2008 04:30 PM	06:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Spanish (Panama)	es_PA	Panama Balboa: PAB	01/02/2008 04:30 PM	06:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Spanish (Peru)	es_PE	Peruvian Nuevo Sol: PEN	02/01/2008 04:30 PM	06:00 AM	1.234,56	Ms. FName LName	Address Line 1, Address Line 2

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
							City, State ZipCode Country
Spanish (Puerto Rico)	es_PR	U.S. Dollar: USD	01-02-2008 04:30 PM	06:00 AM 04:30 PM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Spanish (Paraguay)	es_PY	Paraguayan Guarani: PYG	02/01/2008	06:00 AM 04:30 PM	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Spanish (El Salvador)	es_SVUS	U.S. Dollar: USD	01-02-2008 04:30 PM	06:00 AM 04:30 PM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Spanish (Uruguay)	es_UY	Uruguayan New Peso: UYU	02/01/2008 04:30 PM	06:00 AM 04:30 PM	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Spanish (Venezuela)	es_VE	Venezuelan Bolivar Fuerte: VEF	02/01/2008 04:30 PM	06:00 AM 04:30 PM	1.234,56	Ms. FName LName	Address Line 1, Address Line 2

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
							City, State ZipCode Country
Estonian (Estonia)	et_EE	Estonian Kroon: EEK	2.01.2008 16:30	6:00	1 234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Finnish	fi		2.1.2008 16:30	6:00	1 234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Finnish (Finland)	fi_FI	Euro: EUR	2.1.2008 16:30	6:00	1 234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
French	fr		02/01/2008 16:30	06:00	1 234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
French (Belgium)	fr_BE	Euro: EUR	2/01/2008 16:30	6:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
							City, State ZipCode Country
French (Canada)	fr_CA	Canadian Dollar: CAD	2008-01-02 16:30	06:00	1 234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
French (Switzerland)	fr_CH	Swiss Franc: CHF	02.01.2008 16:30	06:00	1'234.56	Ms. FName LName	Address Line 1, Address Line 2 City Country - State ZipCode
French (France)	fr_FR	Euro: EUR	02/01/2008 16:30	06:00	1 234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
French (Luxembourg)	fr_LU	Euro: EUR	02/01/2008 16:30	06:00	1 234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
French (Monaco)	fr_MC	Moroccan Dirham: MAD	02/01/2008 16:30	06:00	1 234,56	Ms. FName LName	Address Line 1, Address Line 2

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
							City, State ZipCode Country
Croatian (Croatia)	hr_HR	Croatian Kuna: HRK	02.01.2008. 16:30	06:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Hungarian	hu		2008.01.02. 16:30	6:00	1 234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Armenian (Armenia)	hy_AM	Armenian Dram: AMD	1/2/2008 4:30 PM	6:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Indonesian	in		2008/01/02 16:30	06:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Icelandic (Iceland)	is_IS	Iceland Krona: ISK	2.1.2008 16:30	06:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
							City, State ZipCode Country
Italian	it		02/01/2008 16.30	6.00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Italian (Switzerland)	it_CH	Swiss Franc: CHF	02.01.2008 16:30	06:00	1'234.56	Ms. FName LName	Address Line 1, Address Line 2 City Country - State ZipCode
Italian (Italy)	it_IT	Euro: EUR	02/01/2008 16.30	6.00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Hebrew	iw		16:30 02/01/2008	06:00	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Hebrew (Israel)	iw_IL	Israeli Shekel: ILS	16:30 02/01/2008	06:00	1,234.56	Ms. FName LName	Address Line 1, Address Line 2

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
							City, State ZipCode Country
Japanese	ja		2008/01/02 16:30	6:00	1,234.56	LName FName	Country ZipCode State City Address Line 1, Address Line 2
Japanese (Japan)	ja_JP	Japanese Yen: JPY	2008/01/02 16:30	6:00	1,234.56	LName FName	Country ZipCode State City Address Line 1, Address Line 2
Georgian	ka		1/2/2008 4:30 PM	6:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Georgian (Georgia)	ka_GE	Georgian Lari: GEL	1/2/2008 4:30 PM	6:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Kazakh (Kazakhstan)	kk_KZ	Kazakhstan Tenge: KZT	1/2/2008 4:30 PM	6:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
							City, State ZipCode Country
Khmer (Cambodia)	km_KH	Cambodia Riel: KHR	1/2/2008 4:30 PM	6:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Korean	ko		2008. 1. 2 PM 4:30	6:00	1,234.56	LName FName	Country ZipCode State City Address Line 1, Address Line 2
Korean (South Korea)	ko_KR	Korean Won: KRW	2008. 1. 2 PM 4:30	6:00	1,234.56	LName FName	Country ZipCode State City Address Line 1, Address Line 2
Lithuanian (Lithuania)	lt_LT	Lithuanian Lita: LTL	2008.1.2 16.30	06.00	1 234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Latvian (Latvia)	lv_LV	Latvian Lat: LVL	2008.2.1 16:30	06:00	1 234,56	Ms. FName LName	Address Line 1, Address Line 2

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
							City, State ZipCode Country
Malay (Malaysia)	ms_MY	Malaysian Ringgit: MYR	02/01/2008 4:30 PM	6:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Dutch	nl		2-1-2008 16:30	6:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Dutch (Belgium)	nl_BE	Euro: EUR	2/01/2008 16:30	6:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Dutch (Netherlands)	nl_NL	Euro: EUR	2-1-2008 16:30	6:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Dutch (Suriname)	nl_SR	Surinam Dollar: SRD	2-1-2008 16:30	6:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
							City, State ZipCode Country
Norwegian	no		02.01.2008 16:30	06:00	1 234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Norwegian (Norway)	no_NO	Norwegian Krone: NOK	02.01.2008 16:30	06:00	1 234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Polish	pl		2008-01-02 16:30	06:00	1 234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Portuguese	pt		02-01-2008 16:30	6:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Portuguese (Angola)	pt_AO	Angola Kwanza: AOA	02-01-2008 16:30	6:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
							City, State ZipCode Country
Portuguese (Brazil)	pt_BR	Brazilian Real: BRL	02/01/2008 16:30	06:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Portuguese (Portugal)	pt_PT	Euro: EUR	02-01-2008 16:30	6:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Romanian	ro		02.01.2008 16:30	06:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Romanian (Romania)	ro_RO	Romanian Leu (New): RON	02.01.2008 16:30	06:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Russian	ru		02.01.2008 16:30	6:00	1 234,56	Ms. FName LName	Address Line 1, Address Line 2

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
Russian (Russia)	ru_RU	Russian Rouble: RUB	02.01.2008 16:30	6:00	1 234,56	Ms. FName LName	City, State ZipCode Country
Serbian (Latin) (Bosnia and Herzegovina)	sh	Convertible Mark: BAM	02.01.2008. 16:30	06.00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Serbian (Latin) (Serbia and Montenegro)	sh_CS	Serbian Dinar (Serbia): RSD Euro (Montenegro): EUR	02.01.2008. 16:30	06.00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Slovak	sk		2.1.2008 16:30	6:00	1 234,56	Ms. FName LName	Address Line 1, Address Line 2

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
Slovak (Slovakia)	sk_SK	Euro: EUR	2.1.2008 16:30	6:00	1 234,56	Ms. FName LName	City, State ZipCode Country
Slovenian (Slovenia)	sl_SI	Euro: EUR	2.1.08 16:30	6:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Serbian	sr		2.1.2008. 16.30	06.00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Serbian (Bosnia and Herzegovina)	sr_BA	Convertible Mark: BAM	2008-01-02 16:30	06.00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Serbian (Serbia and Montenegro)	sr_CS	Serbian Dinar (Serbia): RSD	2.1.2008. 16.30 Euro (Montenegro): EUR	06.00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
							City, State ZipCode Country
Swedish	sv		2008-01-02 16:30	06:00	1 234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Swedish (Sweden)	sv_SE	Swedish Krona: SEK	2008-01-02 16:30	06:00	1 234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Thai	th		2/1/2008, 16:30 .	6:00 .	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Thai (Thailand)	th_TH	Thai Baht: THB	2/1/2551, 16:30 .	6:00 .	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Tagalog	tl		1/2/2008 4:30 PM	6:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
							City, State ZipCode Country
Tagalog (Philippines)	tl_PH	Philippines Peso: PHP	1/2/2008 4:30 PM	6:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Turkish	tr		02.01.2008 16:30	06:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Ukrainian	uk		02.01.2008 16:30	6:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Ukrainian (Ukraine)	uk_UA	Ukraine Hryvnia: UAH	02.01.2008 16:30	6:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Urdu (Pakistan)	ur_PK	Pakistani Rupee: PKR	1/2/2008 4:30 PM	6:00 AM	1,234.56	Ms. FName LName	Address Line 1, Address Line 2

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
							City, State ZipCode Country
Vietnamese	vi		16:30 02/01/2008	06:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Vietnamese (Vietnam)	vi_VN	Vietnam Dong: VND	16:30 02/01/2008	06:00	1.234,56	Ms. FName LName	Address Line 1, Address Line 2 City, State ZipCode Country
Chinese	zh		2008-1-2 PM4:30	6:00	1,234.56	LName FName	Country ZipCode State City Address Line 1, Address Line 2
Chinese (China)	zh_CN	Chinese Yuan: CNY	2008-1-2 PM4:30	6:00	1,234.56	LName FName	Country ZipCode State City Address Line 1, Address Line 2
Chinese (Hong Kong)	zh_HK	Hong Kong Dollar: HKD	2008 1 2 PM4:30	6:00	1,234.56	LName FName	Country ZipCode State City

Name	Code	Default currency	Date and time format	Time format	Number format	Name format	Address format
							Address Line 1, Address Line 2
Chinese (Taiwan)	zh_TW	Taiwan Dollar: TWD	2008/1/2 PM 4:30	6:00	1,234.56	LName FName	Country ZipCode State City Address Line 1, Address Line 2

Configuring Browsers

Supported Browsers

Database.com supports the following browsers:

Browser	Comments
Windows® Internet Explorer® versions 6, 7, 8, and 9	Salesforce.com strongly recommends using Internet Explorer version 9 over versions 6, 7, and 8. Apply all Microsoft® hotfixes. Internet Explorer 6 is not supported for certain features. Internet Explorer 7 is not supported for Siteforce. The compatibility view feature in Internet Explorer 8 and 9 is not supported in Database.com. For configuration recommendations, see Configuring Internet Explorer on page 48.
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 Configuring Firefox on page 47.
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	Supported on Windows XP and Mac OS X version 10.4 and later. There are no configuration recommendations for Safari.



Note: Database.com uses the *.database.com domain to deliver content. If your users are allowed general access to the Internet, there is no required action. If you whitelist domains, you must add these to your list of allowed domains. If you've disabled third-party cookies (typically enabled by default in all major browsers), you must accept them for Database.com to function properly.



Important: For all browsers you must enable JavaScript, cookies, and SSL 3.0.

Salesforce.com strongly recommends a minimum screen resolution of 1024 x 768 for the best possible user experience.

Configuring Firefox

Salesforce.com recommends using Firefox for best performance and makes every effort to test and support the most recent version.



Tip: Setting `privacy.sanitize.sanitizeOnShutdown` to “True” causes the cache to clear when Firefox shuts down. This increases privacy, but may decrease performance.

To view the contents of your cache, type `about:cache` in the Firefox location bar and press Enter.

Refer to [MozillaZine Knowledge Base](#) and [Firefox Support Home Page](#) for more information on these and other preferences.

Required Settings

The following settings are required:

1. Click **Tools > Options**.
2. Enable JavaScript:
 - a. Go to the Content panel.
 - b. Select the **Enable JavaScript** option.
3. Accept cookies:
 - a. Go to the Privacy panel.
 - b. For the **Firefox will** option, select **Use custom settings for history**.
 - c. Select the **Accept cookies from sites** option.
 - d. Select the **Accept third-party cookies** option.
 - e. For the **Keep until** option, select **they expire**.
4. Set encryption protocols:
 - a. Go to the Advanced panel.
 - b. Click the Encryption tab.
 - c. Select the **Use SSL 3.0** option.
5. Click **OK**.

Advanced Settings

Optionally, configure advanced caching preferences to maximize performance:

1. Type `about:config` in the browser's location bar, and then press Enter.
2. If a warning displays, click **I'll be careful, I promise!**.
3. Search for the following preferences and set them to the recommended value by double-clicking the preference name. Changes take effect immediately.
4. Change how the browser retains common resources across requests by setting the following caching preferences.

Preference	Recommended Value	Default Value
browser.cache.check_doc_frequency	3	3
browser.cache.disk.capacity	50,000 or more; increase to use more hard disk space	50,000
browser.cache.disk.enable	True	True
browser.cache.disk_cache_ssl	True	False
browser.cache.memory.enable	True	True
network.http.use-cache	True	True



Note: You can set some of these preferences by clicking **Tools > Options** in the Firefox browser. Refer to Firefox Help for details.



Tip: Setting `privacy.sanitize.sanitizeOnShutdown` to "True" causes the cache to clear when Firefox shuts down. This increases privacy, but may decrease performance.

To view the contents of your cache, type `about:cache` in the Firefox location bar and press Enter.

Refer to [MozillaZine Knowledge Base](#) and [Firefox Support Home Page](#) for more information on these and other preferences.

Configuring Internet Explorer

Salesforce.com strongly recommends using Internet Explorer version 9 over versions 6, 7, and 8. Apply all Microsoft® hotfixes.

To maximize the performance of Internet Explorer, set the following options in the Internet Options dialog box, which you can open by clicking **Tools > Internet Options**:

General Tab

- From the General tab, click **Settings** under Browsing History (Temporary Internet Files for version 6).
- For the Check for newer versions of stored pages option, select **Automatically**.
- For the Disk space to use option (Amount of disk space to use in version 6), enter at least 50 MB.

Security Tab

- From the Security tab, click **Custom Level** under Internet and scroll to the Scripting section.
- Make sure the Active Scripting option is enabled. JavaScript depends on this setting being enabled.

Privacy Tab

- From the Privacy tab, click **Advanced**.
- Select the **Override automatic cookie handling** option.
- Select the **Always allow session cookies** option.
- For the **Third-party Cookies** option, select **Accept**.

Advanced Tab

From the Advanced tab, scroll to the Security section and do the following:

- Do not select the Do not save encrypted pages to disk option.
- Select the Use SSL 3.0 option.



Tip: The Empty Temporary Internet Files folder when browser is closed option causes the cache to clear when Internet Explorer is shut down. This increases privacy, but may decrease performance.

Customizing Setup Settings

User Permissions Needed	
To modify setup settings:	“Customize Application”

To change your organization's setup settings:

1. Click **Customize > User Interface**.
2. Select or deselect each checkbox to modify the setup settings for your organization.
3. Click **Save**.

Setup Settings

Enable Enhanced Profile List Views

Select this checkbox to activate [enhanced list views](#) and [inline editing](#) on the profiles list page. With inline editing in enhanced profile list views, you can manage multiple profiles at once.

Enable Enhanced Profile User Interface

Select this checkbox to activate the [enhanced profile user interface](#), which allows you to easily navigate, search, and modify settings for a single profile.

Defining a Custom Domain For Your Organization

My Domain Overview

Using My Domain, you can define a custom Database.com domain name for your organization that highlights your brand, or a different term that represents your business. When you sign up for Database.com, a domain name is randomly generated for your organization. You can change the randomly-generated domain name to your own custom domain name only one time. Using a custom domain name provides important advantages, such as increased security and better support for single sign-on. My Domain is also available for test database environments.



Note: My Domain is subject to these additional [Terms of Use](#).

Your domain name uses the standard URL format, including:

- The protocol: https://
- The subdomain prefix: your brand or term

- The domain: database.com

For example, the login URL for a company called Universal Containers would be:
<https://universalcontainers.database.com/>. You can use up to 40 characters.

It's a snap to set up a custom domain name. After you decide on the name or term you want to use, My Domain checks to make sure your subdomain is available. Then it registers the domain name and publishes it to the internet. Log in using the new URL and test the URLs of other pages.

When deployed, you have options for how you want to handle page requests that don't use the new domain name. You can block them entirely or redirect them to the new URL—with or without a message.

 **Important:** After you deploy your new domain name, you can't reverse it. After deployment, all users will be redirected to your new domain.

Setting Up and Rolling Out a Domain Name

When you set up a domain name for your organization, all URLs for Database.com Console pages will change. This table shows you the differences.

URL Type	Old URL	New URL
Login	https://login.database.com	<a href="https://<subdomain>.database.com">https://<subdomain>.database.com
Database.com Console page	<a href="https://na1.database.com/<pageID>">https://na1.database.com/<pageID>	<a href="https://<subdomain>.database.com/<pageID>">https://<subdomain>.database.com/<pageID>



Note: If you implement My Domain in a test database environment, the URL format is
<https://<subdomain>--<testdatabasename>. <instance>.database.com>.

Setting Up Your New Domain Name

- Click **Company Profile > My Domain**.
- Enter the name you want to use within the sample URL. For example, the login URL for a company called Universal Containers would be: <https://universalcontainers.database.com/>. You can use up to 40 characters.
- Click **Check Availability**. If your name is already taken, choose a different one.
- Click **Terms and Conditions** to review your agreement, then select the checkbox.

Testing and Rolling Out Your New Domain Name

Test the new domain name by clicking links within the Database.com Console. You'll notice that all the pages show your new domain name. After you test your domain name, you're ready to roll it out to users.

- When you finish testing your new domain name, click **Company Profile > My Domain** and click **Deploy to Users** to roll out the new domain name to your organization.



Important: After you deploy your new domain name, you can't reverse it. After deployment, all users will be redirected to your new domain.

- Click **Edit** in the My Domain Settings related list. If you want to accept logins from your new domain only, select the **Login Policy** checkbox. If you don't require users to log in from the new domain, they'll be blocked or redirected based on your redirect setting.



Tip: If you block application page requests that don't use the new Database.com domain name URLs, let your users know they need to either update old bookmarks or create new ones for the login page and any links within the Database.com Console. Users will be required to use the new URLs if you block page requests.

- Select a redirect policy. If you want to redirect page requests that don't use the new domain name, select a redirect option. Users can access pages with or without a message explaining the URL change.



Tip: If you choose to redirect page requests to new URLs and provide a warning message, let your users know that they should update their bookmarks the first time they're redirected.

- Click **Save**.



Note: If your domain is registered but has not yet been deployed, URLs will show My Domain URLs when you log in from the My Domain login page.

Guidelines for Implementing Your Domain Name

- After you roll out your new domain name, use My Domain's redirect tools to gradually phase it in. For example, choose the **Redirected with a warning...** option to make sure users update their bookmarks. When your organization is ready to use the new domain URLs exclusively, return to setup and choose the **Blocked** option so users can't use their old URLs.
- If you are using My Domain, you can identify which users are logging in with the new login URL, and when. Click **Manage Users > Login History** and look at the Username and Login URL columns.

Getting System Performance and Maintenance Information

Database.com customers get system performance and maintenance information from `trust.database.com`. Here's how to get that information using your new domain name.

- Go to `trust.database.com`.
- Click the **System Status** tab.
- Enter your domain name to find your instance and check the status.
- Scroll to the System Maintenance table and look for entries for your instance.

Monitoring Your Organization

See Also:

[Monitoring API Usage](#)

Monitoring Apex Usage and Jobs

Monitoring the Apex Job Queue

The Apex job queue lists all Apex jobs that have been submitted for execution. Jobs that have completed execution are listed, as well as those that are not yet finished, including:

- Apex methods with the `future` annotation that have not yet been executed. Such jobs are listed as Future in the Job Type column, and do not have values in the Total Batches or Batches Processed columns.
- Scheduled Apex jobs that have not yet finished executing. Such jobs are listed as Scheduled Apex in the Job Type column, and do not have values in the Total Batches or Batches Processed columns.
- Apex sharing recalculation batch jobs that have not yet finished execution. Such jobs are listed as Sharing Recalculation in the Job Type column. The records in a sharing recalculation job are automatically split into batches. The Total Batches column lists the total number of batches for the job. The Batches Processed column lists the number of batches that have already been processed.
- Batch Apex jobs that have not yet finished execution. Such jobs are listed as Batch Apex in the Job Type column. The records in a batch Apex job are automatically split into batches. The Total Batches column lists the total number of batches for the job. The Batches Processed column lists the number of batches that have already been processed.

 **Note:** Sharing recalculation batch jobs are currently available through a limited release program. For information on enabling Apex sharing recalculation batch jobs for your organization, contact salesforce.com.

The Status column lists the current status of the job. The possible values are:

Status	Description
Queued	Job is awaiting execution
Preparing	The <code>start</code> method of the job has been invoked. This status might last a few minutes depending on the size of the batch of records.
Processing	Job is being processed
Aborted	Job was aborted by a user
Completed	Job completed with or without failures
Failed	Job experienced a system failure

If one or more errors occur during batch processing, the Status Details column gives a short description of the first error. A more detailed description of that error, along with any subsequent errors, is emailed to the user who started the running batch class.

To show a filtered list of items, select a predefined list from the View drop-down list, or click **Create New View** to define your own [custom view](#). This is especially useful if you want to view only `future` methods, or view only Apex batch jobs. To edit or delete any view you created, select it from the View drop-down list and click **Edit**.

Only one batch Apex job's `start` method can run at a time in an organization. Batch jobs that haven't started yet remain in the queue until they're started. Note that this limit doesn't cause any batch job to fail and `execute` methods of batch Apex jobs still run in parallel if more than one job is running.

For any type of Apex job, you can click **Abort Job** in the Action column to stop all processing for that job.

All batch jobs that have completed execution are removed from the batch queue list seven days after completion.

For more information about Apex, see the [Database.com Apex Code Developer's Guide](#).

See Also:

[Monitoring Scheduled Jobs](#)

Monitoring Scheduled Jobs

User Permissions Needed	
To monitor scheduled jobs:	"View Setup and Configuration"

The All Scheduled Jobs page lists all scheduled Apex jobs.

To view this page, click **Monitoring > Scheduled Jobs**. Depending on your permissions, you can perform some or all of the following actions:

- Click **Del** to permanently delete all instances of a scheduled job.
- View the details of a scheduled job, such as:
 - ◊ The name of the scheduled job
 - ◊ The name of the user who submitted the scheduled job
 - ◊ The date and time at which the scheduled job was originally submitted
 - ◊ The date and time at which the scheduled job started
 - ◊ The next date and time at which the scheduled job will run
 - ◊ The type of scheduled job

Monitoring Bulk Data Load Jobs

Monitoring Bulk Data Load Jobs

User Permissions Needed	
To monitor bulk data load jobs:	"Manage Data Integrations"

You can create, update, or delete a large volume of records with the Bulk API, which is optimized for processing large sets of data. It makes it simple to load, update, or delete data from a few thousand to millions of records. Processing a large amount of records takes some time. This page allows you to monitor the progress of current jobs and the results of recent jobs.

Process a set of records by creating a job that contains one or more batches. The job specifies which object is being processed and what type of action is being used (query, insert, upsert, update, or delete). A batch is a set of records sent to the server in an HTTP POST request. Each batch is processed independently by the server, not necessarily in the order it is received.

To track the status of bulk data load jobs that are in progress or recently completed, click **Monitoring > Bulk Data Load Jobs**.

The In Progress Jobs list contains the following columns, shown in alphabetical order:

Column	Description
Job ID	The unique, 15-character ID for this job.

Column	Description
Object	The object type for the data being processed. All data in a job must be of a single object type.
Operation	The processing operation for all the batches in the job. The valid values are: <ul style="list-style-type: none"> delete insert query upsert update hardDelete
Progress	The percentage of batches processed relative to the total number of batches submitted. Progress is not shown when the job is open because the total number of batches in the job is not known until the job is closed. Progress may not accurately reflect the number of records processed. Batches may not all contain the same number of records and they may be processed at different speeds.
Records Processed	The number of records already processed. This number increases as more batches are processed.
Start Time	The date and time when the job was submitted.
Status	The current state of processing for the job. The valid values are: <ul style="list-style-type: none"> Open: The job has been created, and batches can be added to the job. Closed: No new batches can be added to this job. Batches associated with the job may be processed after a job is closed. You cannot edit or save a closed job. Completed: All batches have been processed. Aborted: The job has been aborted. Failed: The job has failed. Batches that were successfully processed in the job cannot be rolled back.
Submitted By	The name of the user that submitted the job.

The Completed Jobs list contains the following columns, shown in alphabetical order. Completed jobs are removed from the list seven days after completion.

Column	Description
End Time	The date and time when the job completed.
Job ID	The unique, 15-character ID for this job.
Object	The object type for the data being processed. All data in a job must be of a single object type.
Operation	The processing operation for all the batches in the job. The valid values are: <ul style="list-style-type: none"> delete insert query upsert update hardDelete

Column	Description
Records Processed	The number of records already processed. This number increases as more batches are processed.
Start Time	The date and time when the job was submitted.
Status	The current state of processing for the job. The valid values are: <ul style="list-style-type: none"> Open: The job has been created, and batches can be added to the job. Closed: No new batches can be added to this job. Batches associated with the job may be processed after a job is closed. You cannot edit or save a closed job. Completed: All batches have been processed. Aborted: The job has been aborted. Failed: The job has failed. Batches that were successfully processed in the job cannot be rolled back.
Submitted By	The name of the user that submitted the job.
Time to Complete	The total time to complete the job.

Viewing Bulk Data Load Job Details

User Permissions Needed
To monitor bulk data load jobs: “Manage Data Integrations”

You can create, update, or delete a large volume of records with the Bulk API, which is optimized for processing large sets of data. It makes it simple to load, update, or delete data from a few thousand to millions of records. Processing a large amount of records takes some time. This page allows you to monitor the progress of current jobs and the results of recent jobs.

To view the details of a bulk data load job:

1. Click **Monitoring > Bulk Data Load Jobs**.
2. Click a **Job ID** link for a job.

The job detail page contains the following fields, shown in alphabetical order:

Field	Description
Apex Processing Time (ms)	The number of milliseconds taken to process triggers and other processes related to the job data. This is the sum of the equivalent times in all batches in the job. This doesn't include the time used for processing asynchronous and batch Apex operations. If there are no triggers, the value is 0.
API Active Processing Time (ms)	The number of milliseconds taken to actively process the job and includes the time tracked in the Apex Processing Time (ms) field, but doesn't include the time the job waited in the queue to be processed or the time required for serialization and deserialization. This is the sum of the equivalent times in all batches in the job.
API Version	The API version for the job.

Field	Description
Completed Batches	The number of batches that have been completed for this job.
Concurrency Mode	The concurrency mode for processing batches. The valid values are: <ul style="list-style-type: none"> <code>parallel</code>: Batches are processed in parallel mode. This is the default value. <code>serial</code>: Batches are processed in serial mode.
Content Type	The content type for the job. The valid values are: <ul style="list-style-type: none"> <code>CSV</code>—data in CSV format <code>XML</code>—data in XML format (default option) <code>ZIP_CSV</code>—data in CSV format in a zip file containing binary attachments <code>ZIP_XML</code>—data in XML format in a zip file containing binary attachments
End Time	The date and time when the job completed.
External ID Field	The name of the external ID field for an <code>upsert()</code> .
Failed Batches	The number of batches that have failed for this job.
Job ID	The unique, 15-character ID for this job.
In Progress Batches	The number of batches that are in progress for this job.
Object	The object type for the data being processed. All data in a job must be of a single object type.
Operations	The processing operation for all the batches in the job. The valid values are: <ul style="list-style-type: none"> <code>delete</code> <code>insert</code> <code>query</code> <code>upsert</code> <code>update</code> <code>hardDelete</code>
Progress	The percentage of batches processed relative to the total number of batches submitted. Progress is not shown when the job is open because the total number of batches in the job is not known until the job is closed. Progress may not accurately reflect the number of records processed. Batches may not all contain the same number of records and they may be processed at different speeds.
Queued Batches	The number of batches queued for this job.
Records Failed	The number of records that were not processed successfully in this job.
Records Processed	The number of records processed at the time the request was sent. This number increases as more batches are processed.

Field	Description
Retries	The number of times that Database.com attempted to save the results of an operation. The repeated attempts are due to a problem, such as a lock contention.
Start Time	The date and time when the job was submitted.
Status	The current state of processing for the job. The valid values are: <ul style="list-style-type: none"> Open: The job has been created, and batches can be added to the job. Closed: No new batches can be added to this job. Batches associated with the job may be processed after a job is closed. You cannot edit or save a closed job. Completed: All batches have been processed. Aborted: The job has been aborted. Failed: The job has failed. Batches that were successfully processed in the job cannot be rolled back.
Submitted By	The name of the user that submitted the job.
Time to Complete	The total time to complete the job.
Total Processing Time (ms)	The number of milliseconds taken to process the job. This is the sum of the total processing times for all batches in the job.

The job detail page includes a related list of all the batches for the job. The related list provides **View Request** and **View Response** links for each batch. If the batch is a CSV file, the links return the request or response in CSV format. If the batch is an XML file, the links return the request or response in XML format. These links are available for batches created in API version 19.0 and later.

The batch related list contains the following fields, shown in alphabetical order:

Field	Description
Apex Processing Time (ms)	The number of milliseconds taken to process triggers and other processes related to the batch data. If there are no triggers, the value is 0. This doesn't include the time used for processing asynchronous and batch Apex operations.
API Active Processing Time (ms)	The number of milliseconds taken to actively process the batch, and includes Apex processing time. This doesn't include the time the batch waited in the queue to be processed or the time required for serialization and deserialization.
Batch ID	The ID of the batch. May be globally unique, but does not have to be.
End Time	The date and time in the UTC time zone that processing ended. This is only valid when the state is Completed.
Records Failed	The number of records that were not processed successfully in this batch.
Records Processed	The number of records processed in this batch at the time the request was sent. This number increases as more batches are processed.

Field	Description
Retry Count	The number of times that Database.com attempted to save the results of an operation. The repeated attempts are due to a problem, such as lock contention or a batch taking too long to process.
Start Time	The date and time in the UTC time zone when the batch was created. This is not the time processing began, but the time the batch was added to the job.
State Message	Contains the reasons for failure if the batch didn't complete successfully.
Status	<p>The current state of processing for the batch:</p> <ul style="list-style-type: none"> Queued: Processing of the batch has not started yet. If the job associated with this batch is aborted, this batch isn't processed and its state is set to Not Processed. In Progress: The batch is currently being processed. If the job associated with this batch is aborted, this batch is still processed to completion. Completed: The batch has been processed completely and the result resource is available. The result resource indicates if some records have failed. A batch can be completed even if some or all the records have failed. If a subset of records failed, the successful records aren't rolled back. Failed: The batch failed to process the full request due to an unexpected error, such as the request being compressed with an unsupported format, or an internal server error. Not Processed: The batch failed to process the full request due to an unexpected error, such as the request being compressed with an unsupported format, or an internal server error.
Total Processing Time (ms)	The number of milliseconds taken to process the batch. This excludes the time the batch waited in the queue to be processed.
View Request	Click the link for a batch to see the request.
View Result	Click the link for a batch to see the results.

Monitoring Debug Logs

Monitoring Debug Logs

User Permissions Needed	
To view, retain, and delete debug logs:	“Manage Users”

You can retain and manage the debug logs for specific users.

Transactions can be generated from the following:

- API
- executeanonymous calls
- Web services

To view saved debug logs, click **Monitoring > Debug Logs**.

From this page, click **New** to specify a user that you want to retain debug logs for.

After you have specified a user or users to retain debug logs for, you can:

- Click **Delete** to stop retaining debug logs for a specific user.
- Click **Reset** to reset the number of debug logs for a particular user.
- Click **Filters** to specify what gets logged for that user, as well as the amount of information. See [Setting Debug Log Filters](#).

After you have started retaining debug logs, you can:

- Click **View** to [view a specific log's details](#).
- Click **Download** to download the debug log as an XML file.

Viewing Debug Logs

User Permissions Needed	
To use the Developer Console:	“View All Data”
To use the execute anonymous text entry box:	“Author Apex”
To save changes to Apex classes and triggers:	“Author Apex”

You can retain and manage the debug logs for specific users.

To view the details of a debug log, click **Monitoring > Debug Logs**, and then click **View** next to the debug log you want to examine. Click **Download** to download the log as an XML file.

The debug log contains information about the transaction, such as if it was successful, the size of the log (in bytes), how long the transaction took in milliseconds, and so on. The log itself contains additional information about the transaction, depending on the filters set for the user.

Inspecting the Debug Log Sections

After you generate a debug log, the type and amount of information listed depends on the [filter values](#) you set for the user. However, the format for a debug log is always the same.

A debug log has the following sections:

Header

The header contains the following information:

- The version of the API used during the transaction.
- The [log category and level](#) used to generate the log. For example:

The following is an example of a header:

```
22.0
APEX_CODE,DEBUG;APEX_PROFILING,INFO;CALLOUT,INFO;DB,INFO;SYSTEM,DEBUG;VALIDATION,INFO;VISUALFORCE,INFO;
WORKFLOW,INFO
```

In this example, the API version is 22.0, and the following debug log categories and levels have been set:

Apex Code	DEBUG
Apex Profiling	INFO

Callout	INFO
Database	INFO
System	DEBUG
Validation	INFO
Visualforce	INFO
Workflow	INFO

Execution Units

An execution unit is equivalent to a transaction. It contains everything that occurred within the transaction. The execution is delimited by `EXECUTION_STARTED` and `EXECUTION_FINISHED`.

Code Units

A code unit is a discrete unit of work within a transaction. For example, a trigger is one unit of code, as is a `webService` method, or a validation rule.



Note: A class is **not** a discrete unit of code.

Units of code are indicated by `CODE_UNIT_STARTED` and `CODE_UNIT_FINISHED`. Units of work can embed other units of work. For example:

```
EXECUTION_STARTED
CODE_UNIT_STARTED|[EXTERNAL]execute_anonymous_apex
CODE_UNIT_STARTED|[EXTERNAL]MyTrigger on Merchandise trigger event BeforeInsert for
[new]
CODE_UNIT_FINISHED <-- The trigger ends
CODE_UNIT_FINISHED <-- The executeAnonymous ends
EXECUTION_FINISHED
```

Units of code include, but are not limited to, the following:

- Triggers
- Workflow invocations and time-based workflow
- Validation rules
- `@future` method invocations
- Web service invocations
- `executeAnonymous` calls
- Execution of the batch Apex `start` and `finish` methods, as well as each execution of the `execute` method
- Execution of the Apex `System.Schedule.execute` method

Log Lines

Included inside the units of code. These indicate what code or rules are being executed, or messages being specifically written to the debug log. For example:

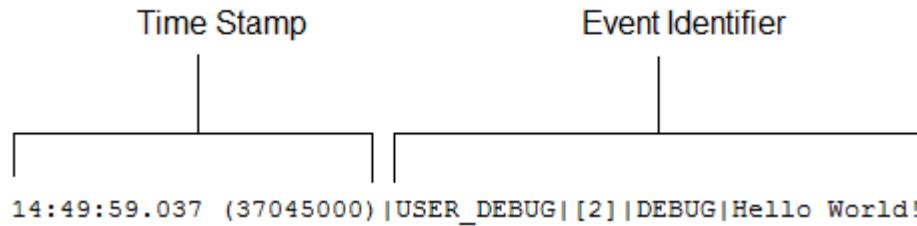


Figure 1: Debug Log Line Example

Log lines are made up of a set of fields, delimited by a pipe (|). The format is:

- *timestamp*: consists of the time when the event occurred and a value between parentheses. The time is in the user's time zone and in the format *HH:mm:ss.sss*. The value represents the time elapsed in nanoseconds since the start of the request. The elapsed time value is excluded from logs reviewed in the Developer Console.
- *event identifier*: consists of the specific event that triggered the debug log being written to, such as `SAVEPOINT_RESET` or `VALIDATION_RULE`, and any additional information logged with that event, such as the method name or the line and character number where the code was executed.

Additional Log Data

In addition, the log contains the following information:

- Cumulative resource usage—Logged at the end of many code units, such as triggers, `executeAnonymous`, batch Apex message processing, `@future` methods, Apex test methods, Apex web service methods.
- Cumulative profiling information—Logged once at the end of the transaction. Contains information about the most expensive queries (that used the most resources), DML invocations, and so on.

The following is an example debug log:

```
23.0
APEX_CODE,DEBUG;APEX_PROFILING,INFO;CALLOUT,INFO;DB,INFO;SYSTEM,DEBUG;VALIDATION,INFO;VISUALFORCE,INFO;
WORKFLOW,INFO
11:47:46.030 (30064000) |EXECUTION_STARTED
11:47:46.030 (30159000) |CODE_UNIT_STARTED|[EXTERNAL]|TRIGGERS
11:47:46.030 (30271000) |CODE_UNIT_STARTED|[EXTERNAL]|01qD00000004JvP|myTrigger on Merchandise
trigger event BeforeUpdate for [001D000000IzMaE]
11:47:46.038 (38296000) |SYSTEM_METHOD_ENTRY|[2]|System.debug(ANY)
11:47:46.038 (38450000) |USER_DEBUG|[2]|DEBUG|Hello World!
11:47:46.038 (38520000) |SYSTEM_METHOD_EXIT|[2]|System.debug(ANY)
11:47:46.546 (38587000) |CUMULATIVE_LIMIT_USAGE
11:47:46.546|LIMIT_USAGE_FOR_NS|(default)|
    Number of SOQL queries: 0 out of 100
    Number of query rows: 0 out of 50000
    Number of SOSL queries: 0 out of 20
    Number of DML statements: 0 out of 150
    Number of DML rows: 0 out of 10000
    Number of script statements: 1 out of 200000
    Maximum heap size: 0 out of 6000000
    Number of callouts: 0 out of 10
    Number of Email Invocations: 0 out of 10
    Number of fields describes: 0 out of 100
    Number of record type describes: 0 out of 100
    Number of child relationships describes: 0 out of 100
    Number of picklist describes: 0 out of 100
    Number of future calls: 0 out of 10

11:47:46.546|CUMULATIVE_LIMIT_USAGE_END
```

```
11:47:46.038 (38715000) |CODE_UNIT_FINISHED|myTrigger on Merchandise trigger event BeforeUpdate
for [001D000000IzMaE]
11:47:47.154 (1154831000) |CODE_UNIT_FINISHED|TRIGGERS
11:47:47.154 (1154881000) |EXECUTION_FINISHED
```

 **Note:** Visualforce isn't available in Database.com.

See Also:

[Monitoring Debug Logs](#)

Monitoring the Workflow Queue

User Permissions Needed	
To manage the workflow queue:	“Modify All Data”

When a workflow rule that has time-dependent actions is triggered, use the workflow queue to view pending actions and cancel them if necessary.

To view pending actions:

1. Click **Monitoring > Time-Based Workflow**.
2. Click **Search** to view all pending actions for any active workflow rules, or set the filter criteria and click **Search** to view only the pending actions that match the criteria. The filter options are:
 - **Workflow Rule Name:** The name of the workflow rule.
 - **Object:** The object that triggered the workflow rule. Enter the object name in the singular form.
 - **Scheduled Date:** The date the pending actions are scheduled to occur.
 - **Create Date:** The date the record that triggered the workflow was created.
 - **Created By:** The user who created the record that triggered the workflow rule.
 - **Record Name:** The name of the record that triggered the workflow rule.

The filter is not case-sensitive. For more information on filters, see [Entering Filter Criteria](#) on page 81.

To cancel pending actions:

1. Select the box next to the pending actions you want to cancel. Optionally, check the box in the column header to select all currently displayed items.
2. Click **Delete**.

Monitoring Resources

User Permissions Needed	
To view storage usage:	“Manage Users”

Storage is divided into two categories: data storage and file storage. File storage includes files imported through the API while data storage includes custom objects.

Data storage and file storage are calculated asynchronously, so if you import or add a large number of files, your organization's storage usage will not be updated immediately.

Storage Capacity

Database.com includes a minimum amount of data storage and file storage for free. The minimum allotted to a Database.com organization is 1 GB of file storage and 512 MB of data storage, or up to 100,000 records.

If your organization uses custom user licenses, contact salesforce.com to determine if these licenses provide additional storage. For a description of user licenses, see [Understanding User License Types](#) on page 197.

Viewing Storage Usage

To view your organization's current storage usage click **Data Management > Storage Usage**. You can view the available space for data storage and file storage, the amount of storage in use per record type, the top users according to storage utilization, and the largest files in order of size. To view what types of data a particular user is storing, click that user's name.

Administrators can view storage usage on a user by user basis:

1. Click **Manage Users > Users**.
2. Click the name of any user.
3. Click **View** next to the **Used Data Space** or **Used File Space** fields to view that user's storage usage by record type.

Individual users can view their own storage usage in their personal information (see [Editing Your Personal Information](#) on page 74).

Increasing Storage

When your organization has reached its storage limit, you will not be able to create any new data or upload new files.

To increase your storage limit, you can purchase additional storage space. Contact salesforce.com to increase your storage limit.

Monitoring Background Jobs

User Permissions Needed	
To monitor background jobs:	“View Setup and Configuration”

To view any background jobs in your organization (such as sharing rule recalculation jobs), select **Monitoring > Background Jobs**. The All Background Jobs page shows the details of background jobs.

You can only monitor background jobs on this page. To abort a job, contact salesforce.com.

Monitoring Metadata Deployments

User Permissions Needed

To view metadata deployments:

“Modify All Data”

You can use the Metadata API to deploy XML file representations of components into an organization. A component is an instance of a metadata type in the Metadata API. For example, CustomObject is a metadata type for custom objects, and the MyCustomObject__c component is an instance of a custom object.

The easiest way to access the Metadata API is to use the Force.com IDE or Force.com Migration Tool. These tools are built on top of the Metadata API and use the standard Eclipse and Ant tools respectively to simplify the task of working with the Metadata API.

The deploy call in the Metadata API is asynchronous and may take a while to complete. The Force.com IDE and Force.com Migration Tool use the deploy call to move XML file representations of components into an organization. To track the status of deployments that are in progress or completed in the last 24 hours for these tools or other clients that are deploying metadata, click **Deploy > Monitor Deployments**.

The Deployments In Progress list contains the following columns:

Column	Description
Created By	The name of the user performing the deployment.
Start Time	The date and time when the deployment started.
Validate Only	Indicates whether this deployment is being used to check the validity of the deployed files without making any changes in the organization. A validate-only deployment does not deploy any components or change the organization in any way.
Status	The following values are possible: <ul style="list-style-type: none"> Queued—The deployment has not started. It is waiting in a queue. In Progress—The deployment has started, but has not completed yet. Processing Type: <i>type</i>—The deployment has started, and is processing a metadata type, where <i>type</i> is a metadata type, such as <i>CustomObject</i>. Processing <i>name</i>—The deployment has started, and is processing a component, where <i>name</i> is a component name, such as <i>MyCustomObject__c</i>. Running Test <i>ApexClass.testName</i>—An Apex test class is running, where <i>ApexClass</i> is the Apex class name and <i>testName</i> is the name of the test.
Components	The progress of the deployment. For example, <i>10 / 15 (3 errors)</i> indicates that ten components have been processed successfully out of a total of 15. There were errors for three other components processed so far.
Tests	The progress of the Apex tests that have been run. For example, <i>13 / 20 (2 errors)</i> indicates that 13 tests have been run successfully out of a total of 20. There were errors for two other tests that have run so far. The value for the total number of tests is not accurate until the test phase of the deployment has started.

The Completed Deployments Last 24 Hours list contains the following columns. Completed deployments are removed from the list 24 hours after completion.

Column	Description
Created By	The name of the user who performed the deployment.
Start Time	The date and time when the deployment started.
End Time	The date and time when the deployment completed.
Validate Only	Indicates whether this deployment is being used to check the validity of the deployed files without making any changes in the organization. A validate-only deployment does not deploy any components or change the organization in any way.
Status	The following values are possible: <ul style="list-style-type: none"> Completed—The deployment completed successfully. Completed with Errors—The deployment completed, but some components failed to deploy or some tests failed. This status cannot happen in a production organization as any errors cause the deployment to be rolled back. Failed—The deployment had some errors and no changes were made to the organization.
Components	The number of components, including those with errors, that were processed in the deployment. For example, 10 (1 error) indicates that ten components were processed and there was an error for one component.
Tests	The number of Apex tests, including those with errors, that were run. For example, 13 (2 errors) indicates that 13 tests were run and there were errors for two tests.

Monitoring Login History

User Permissions Needed	
To monitor logins:	“Manage Users”

On this page, Administrators can monitor the successful and failed login attempts for their organization. The columns on this page provide information about each login attempt. The login history page displays the most recent 20,000 entries in the login history database. If you need to see more records, you can download the information to a CSV or GZIP file.

Downloading Login History

To download the information into a CSV or GZIP file:

1. Click **Manage Users > Login History**.
2. Click one of the following radio buttons:
 - **Excel csv file:** This downloads a CSV file of all user logins to your Database.com organization for the past six months. This report includes logins through the API.
 - **gzipped Excel csv file:** This downloads a CSV file of all user logins to your Database.com organization for the past six months. This report includes logins through the API. The file is compressed and this is the preferred option for quickest download time.
3. Select the file contents. You can choose All Logins, IE6 Logins Only, or Web Site Logins Only.
4. Click **Download Now**.



Note: Older versions of Microsoft Excel cannot open files with more than 65,536 rows. If you cannot open a large file in Excel, see the [Microsoft Help and Support article about handling large files](#).

You can create new list views sorted by Login Time and Login URL. For example, you could create a view of all logins between a particular time range. Like the default view, a custom view filters the most recent 20,000 rows in the login history database.

To create a new view, click **Create New View** from the page. Fill in the following:

1. Enter View Name

Enter the name to appear in the View drop-down list.

2. Specify Filter Criteria.

You can filter by Login Time and Login URL.

3. Select Fields to Display

The default fields are automatically selected. You can choose up to 15 different fields to display in your view. You can display only the fields that are available in your page layout. When you select a long text area field, up to 255 characters are displayed in the list view.

To see the last ten successful and failed logins for a specific user, navigate to **Manage Users > Users**, click on the Full Name for the user, and scroll to the Login History section.

Single Sign-On with SAML

If your organization has set up single sign-on using identity provider certificates (written in SAML), you may see login history messages [specific to single sign-on](#).

My Domain

If you are using My Domain, you can identify which users are logging in with the new login URL, and when. Click **Manage Users > Login History** and look at the Username and Login URL columns.

Monitoring Training History

User Permissions Needed	
To view training history:	“Manage Users”

As an administrator, it is important to know that your team is learning how to use Database.com effectively. The Training Class History shows you all of the Database.com training classes your users have taken.

Administrators can view the Training Class History by choosing **Manage Users > Training History**. After taking a live training class, users must submit the online training feedback form to have their training attendance recorded in the training history.

Managing Custom Settings

Custom Settings Overview

User Permissions Needed	
To manage, create, edit, and delete custom settings:	“Customize Application”

Custom settings are similar to custom objects and enable application developers to create custom sets of data, as well as create and associate custom data for an organization, profile, or specific user. All custom settings data is exposed in the application cache, which enables efficient access without the cost of repeated queries to the database. This data can then be used by formula fields, validation rules, Apex, and the Web services API.

There are two types of custom settings:

List Custom Settings

A type of custom setting that provides a reusable set of static data that can be accessed across your organization. If you use a particular set of data frequently within your application, putting that data in a list custom setting streamlines access to it. Data in list settings does not vary with profile or user, but is available organization-wide. Examples of list data include two-letter state abbreviations, international dialing prefixes, and catalog numbers for products. Because the data is cached, access is low-cost and efficient: you don't have to use SOQL queries that count against your governor limits.

Hierarchy Custom Settings

A type of custom setting that uses a built-in hierarchical logic that lets you “personalize” settings for specific profiles or users. The hierarchy logic checks the organization, profile, and user settings for the current user and returns the most specific, or “lowest,” value. In the hierarchy, settings for an organization are overridden by profile settings, which, in turn, are overridden by user settings.

The following examples illustrate how you can use custom settings:

- A shipping application requires users to fill in the country codes for international deliveries. By creating a list setting of all country codes, users have quick access to this data without needing to query the database.
- An application calculates and tracks compensation for its sales reps, but commission percentages are based on seniority. By creating a hierarchy setting, the administrator can associate a different commission percentage for each profile in the sales organization. Within the application, one formula field can then be used to correctly calculate compensation for all users; the personalized settings at the profile level inserts the correct commission percentage.
- An application displays a map of account locations, the best route to take, and traffic conditions. This information is useful for sales reps, but account executives only want to see account locations. By creating a hierarchy setting with custom checkbox fields for route and traffic, you can enable this data for just the “Sales Rep” profile.

Follow these steps to create and use custom settings:

1. [Create the custom setting.](#)
2. [Add fields to the custom setting.](#)
3. [Add data and set the access level for the custom setting data.](#)
4. [Reference the custom setting data in your application, using formula fields, validation rules, Apex, or the Web services API.](#)

Managing Custom Settings

Click **New** to [create a new custom setting](#). After you create a custom setting, you must [add fields](#) to it.

After you create a custom setting, you can do any of the following:

- Click **Edit** next to the name of a custom setting to change the name, label, or description of a custom setting.

- Click **Del** to delete a custom setting.
- Click **Manage** to [add data](#) to a custom setting. You should [add fields](#) before you add data.

Accessing Custom Settings

User Permissions Needed	
To manage, create, edit, and delete custom settings:	“Customize Application”

You can access custom settings from formula fields, validation rules, Apex, and the Web services API. Some sample code segments are provided below.

Formula Fields

Formula fields only work for hierarchy custom settings; they can't be used for list custom settings..

```
{ !$Setup.CustomSettingName__c.CustomFieldName__c }
```

Apex

Apex can access both custom setting types. For more information on all the custom setting methods and Apex, see the [Database.com Apex Code Developer's Guide](#).

Samples for List Custom Settings

When you add data to a custom setting, you must name each set of data. Then you can distinguish between the sets of data by the data set name. The following returns a map of custom settings data. The `getAll` method returns values for all custom fields associated with the list setting.

```
Map<String dataset_name, CustomSettingName__c> mcs = CustomSettingName__c.getAll();
```

The following example uses the `getValues` method to return all the field values associated with the specified data set. This method can be used with both list and hierarchy custom settings, using different parameters.

```
CustomSettingName__c mc = CustomSettingName__c.getValues(data_set_name);
```

Samples for Hierarchy Custom Settings

The following example uses the `getOrgDefaults` method to return the data set values for the organization level:

```
CustomSettingName__c mc = CustomSettingName__c.getOrgDefaults();
```

The following example uses the `getInstance` method to return the data set values for the specified profile. The `getInstance` method can also be used with a user Id.

```
CustomSettingName__c mc = CustomSettingName__c.getInstance(Profile_ID);
```

Web Services API

Custom settings that have **Privacy** defined as **Public** are exposed to the API in the same way custom objects are exposed. Use any tool with API access to perform query or profile-permission-setting operations. For more information, see the [Web Services API Developer's Guide](#).



Note: You can also access custom settings data through a Standard Object Query Language (SOQL) query but this method doesn't make use of the application cache. It's similar to querying a custom object.

Adding Custom Settings Data

User Permissions Needed	
To manage, create, edit, and delete custom settings:	“Customize Application”

After you [define your custom settings](#) and [add fields](#), you need to populate the fields with data.

You can define one or more data sets. For list custom settings, each data set is named and can be accessed by that name using Apex, formula fields, and so on.

For custom settings that are hierarchies, the data is accessed based on the access level (user, profile, or organization). The lowest level is used first, which means if you defined a data set at the user level, unless otherwise specified in your application, that data is used. For example, you might want to specify different contact numbers for your application: one for the general user, and one that is only displayed for system administrators.

To add data to custom setting fields:

1. Click **Develop > Custom Settings**, then click **Manage** next to a custom setting. Or from the detail page for a custom setting, click **Manage**.
2. Click **New** or **Edit** next to an existing data set.
3. Add or change data.

For custom settings that are lists:

- a. Specify or change the name for the data set. This name is used by Apex, formula fields, and so on.
- b. Enter or change data for all fields.
- c. Click **Save**.

For custom settings that are hierarchies:

- a. For the default organization level values, enter or change the data for the fields. The default organization location is automatically populated.
- b. For profile or user level values, select either **Profile** or **User** from the **Location** picklist. Enter the name of the profile or user, or use the lookup dialog search. Then enter or change the data for the fields.
- c. Click **Save**.

Adding Custom Settings Fields

User Permissions Needed

To manage, create, edit, and delete custom settings: “Customize Application”

After you [define custom settings](#), you need to add custom fields to them. The custom fields contain the data used by the custom setting.

To add custom fields to a custom setting:

1. Click **Develop > Custom Settings**.
2. Click the name of the custom setting that you want to add fields to. (If you just created a custom setting, you are taken directly to the Custom Setting Detail page.)
3. Click **New**.
4. Select a field type and click **Next**.
5. [Enter the details](#) for your custom field.
6. Once you confirm the information, click **Save** or **Save & New**.

After you add the required fields, you need to [add data](#), and for hierarchy custom settings, specify the access level.

Defining Custom Settings

User Permissions Needed

To manage, create, edit, and delete custom settings: “Customize Application”

To create or edit a custom setting:

1. Click **Develop > Custom Settings**.
2. Click **New** to create a new custom setting, click **Edit** next to the name of a custom setting, or click **Edit** while viewing the details of a custom setting.
3. Define the following:
 - **Label**—Enter the label displayed in the application.
 - **Object Name**—Enter the name to be used when the custom setting is referenced by formula fields, validation rules, Apex, or the Web services API.



Note: Salesforce.com recommends using ASCII for the Object Name. The name can't exceed 38 ASCII characters. If you use double byte, there are additional limits on the number of characters allowed.

- **Setting Type**—Select a type of List or Hierarchy. The List type defines application-level data, such as country codes or state abbreviations. The Hierarchy type defines personalization settings, such as default field values, that can be overridden at lower levels in the hierarchy.



Important: After you save a custom setting, you cannot change this value.

- **Visibility**—Select a visibility of Protected or Public.
 - ◊ **Protected**—The custom setting controls app behavior that you, the developer of the application, want to control.

- ◊ Public—The custom setting controls app behavior that you, the developer of the application, want to allow the users to control.



Important: After you save a custom setting, you cannot change this value.

4. Enter an optional description of the custom setting. A meaningful description will help you remember the differences between your custom settings when you're viewing them in a list.
5. Click **Save**.

After you create a custom setting, you must also [add fields](#) to the custom setting.

Viewing Custom Settings

User Permissions Needed	
To manage, create, edit, and delete custom settings:	“Customize Application”

After you create a custom setting, you can view the details of the custom setting, manage the custom setting, and add fields.

Click **Develop > Custom Settings**, then click the name of the custom setting you'd like to view. While viewing a custom setting, you can:

- Click **Edit** to make changes to a custom setting.
- Click **Delete** to delete a custom setting.
- Click **Manage** to [add data](#) to a custom setting.

In addition, click **New** to [add fields](#) to the custom setting.

Custom Settings Limits

User Permissions Needed	
To manage, create, edit, and delete custom settings:	“Customize Application”

Database.com imposes these limits on the amount of cached data and on custom settings:

- The total amount of cached data allowed for your organization is the lesser of 10MB or 1 MB multiplied by the number of Database.com Admin user licenses in your organization.
- 300 fields per custom setting.
- You can't share a custom setting object or record.
- No owner is assigned when a custom setting is created, so the owner can't be changed.
- Custom settings are a type of custom object. Each custom setting counts against the total number of custom objects available for your organization.

To see how much custom settings data your organization is using, click **Develop > Custom Settings**. This page also details how much resource each custom setting uses, including the number of records and the size of the custom setting definition.

Managing Custom Settings Data

User Permissions Needed

To manage, create, edit, and delete custom settings: “Customize Application”

After defining [custom settings](#) and [adding fields](#), populate the fields:

1. Click **Develop > Custom Settings**.
2. Click **Manage** next to a custom setting, or from the detail page for a custom setting.
3. Provide or change values for the custom setting.
 - If you are managing a list setting:
 - ◊ Click **New** to add data to the fields.
 - ◊ Click **Edit** next to the name of an existing set of data to change the name of the data set or to change the data.
 - ◊ Click **Del** next to the name of an existing set of data to delete the data set.
 - If you are managing a hierarchy setting, decide where in the permission hierarchy you want to add default data (organization, profile, or user).

To add default data at the organization level, click **New** in the Default Organization Level Value section. If data has already been defined for the organization, you can only edit or delete it.

To add default data at the profile or user level, click **New** in the lower section of the page, near the Setup Owner.

After you have defined data, you can:

- Click **Edit** in the Default Organization Level Value section to change the default data set at the organization level, or **Delete** to delete it (this is only for hierarchical custom settings.)
- Click **View** next to the name of an existing set of data to view the data (this is only for hierarchical custom settings.)
- Click **Edit** next to the name of an existing set of data to change the name of the data set or to change the data.
- Click **Del** next to the name of an existing set of data to delete the data set.

Viewing Custom Settings Data

User Permissions Needed

To manage, create, edit, and delete custom settings: “Customize Application”

After you [add fields](#) and [add data](#) to those fields, you can view the data.

1. Click **Develop > Custom Settings**, then click **Manage** next to a custom setting that has already been defined. Or from the detail page for a custom setting, click **Manage**.
2. Click **View** next to the data set you want to view (this is only for hierarchical custom settings).

Critical Updates Overview

User Permissions Needed	
To view critical updates:	“View Setup”
To activate critical updates:	“Modify All Data” and “Customize Application”

Salesforce.com periodically releases updates that improve the performance, logic, and usability of Database.com, but may affect your existing customizations. When these updates become available, Database.com lists them at **Critical Updates** and sends a notification email to administrators.

To ensure a smooth transition, each update has an opt-in period during which you can manually activate and deactivate the update an unlimited number of times to evaluate its impact on your organization and modify affected customizations as necessary. The opt-in period ends on the auto-activation date, at which time Database.com permanently activates the update.



Caution: Salesforce.com recommends testing each update by activating it in either your **QA Database** or your production environment during off-peak hours.

To manage critical updates, click **Critical Updates**. From this page, you can:

- View the summary, status, and auto-activation date for any update that Database.com has not permanently activated.
- Click **Review** to view the detail page of any update that Database.com has not permanently activated. The details include a list of the customizations in your organization that the update might affect and the activation history, which lists each time the update was activated and deactivated.
- Click **Activate** to activate any inactive update.
- Click **Deactivate** to deactivate any active update that Database.com has not permanently activated.

Notes on Critical Updates

- Database.com analyzes your organization to determine if a critical update potentially affects your customizations. If your customizations are not affected, Database.com automatically activates the update in your organization, and the update does not appear when you click **Critical Updates**.
- On the scheduled auto-activation date, Database.com permanently activates the update. After auto-activation, you cannot deactivate the update.

Personalizing Database.com

Personal Setup Overview

The Personal Setup page, accessed by clicking **Personal Setup**, contains setup and customization options to help you personalize the Database.com console for your personal use.

The sidebar includes tools for [browsing and searching setup options](#).

My Personal Information

Expand the **My Personal Information** folder under **Personal Setup** to access the following options.

- **Personal Information**—Edit your user information, view login history, and more (see [Editing Your Personal Information](#) on page 74).
- **Change My Password**—Change your password (see [Changing Your Password](#) on page 77).
- **Reset My Security Token**—Reset the security token that you may need to log into Database.com from outside your company's trusted network (see [Resetting Your Security Token](#) on page 79).
- **Grant Login Access**—Allow salesforce.com Customer Support representatives or your administrator to log in to your account (see [Granting Login Access](#) on page 79).

Managing Your Personal Information

Editing Your Personal Information

To update your personal information, click **My Personal Information > Personal Information**. Administrators can edit any user's information as described in [Editing Users](#) on page 196.

From the personal information page, you can change the following:

- **Personal Information**—To make changes, click **Edit**.

See [User Fields](#) on page 74 for a list of the user fields.

If you change your email address, a confirmation message will be sent to the new address. You must click the link provided in that message for the new email address to take effect. This process is to ensure system security.

- **Change Password**—See [Changing Your Password](#) on page 77.
- **Storage Space**—Click **View** next to the **Used Data Space** or **Used File Space** field to see how much storage space you are using. For information on organization storage limits, see [Monitoring Resources](#) on page 62.
- **Login History**—Administrators can view the user's last ten successful and failed login attempts.

You also have access to view the following:

- **Public Group Membership**—Lists the public groups to which you belong, as determined by your administrator.
- **Managers in the Role Hierarchy**—Lists the users above you in the role hierarchy, as defined by your administrator. See [Managers in the Role Hierarchy](#) on page 207.
- **Remote Access**—Lists the [remote access applications](#) that you have granted access to.

User Fields

A user's personal information—or user detail—page has the following fields, listed in alphabetical order. Some of these fields may not be visible or editable depending on your permissions.

Field	Description
Accessibility Mode	Checkbox that enables or disables a user interface mode designed for visually-impaired users. See “Enabling Accessibility Mode” in the online help.
Active	Administrative checkbox that enables or disables user login to the service.
Address	Street address for user. Up to 255 characters are allowed in this field.

Field	Description
Admin newsletter	Opt in to receive administrator-targeted promotional emails from salesforce.com. This field is not available if your organization has disabled your choice to receive emails from salesforce.com.
Alias	Short name to identify user on pages where the entire name does not fit. Up to eight characters are allowed in this field.
Api Token	Indicates whether or not an API token has ever been reset. Salesforce.com uses this field to help you troubleshoot issues related to API tokens if issues should occur.
Checkout Enabled	Indicates whether the user is notified by email when his or her Checkout account is activated and available for login. You must have “Manage Billing” permission to enable.
City	City portion of user’s address. Up to 40 characters are allowed in this field.
Color-Blind Palette on Charts	Indicates whether the option to set an alternate color palette for charts has been enabled. The alternate palette has been optimized for use by color-blind users.
Company	Company name where user works. Up to 40 characters are allowed in this field.
Country	Country portion of user’s address. Up to 40 characters are allowed in this field.
Created By	User who created the user including creation date and time. (Read only)
Currency	User’s default currency. Shown only in organizations using multiple currencies. This must be one of the active currencies for the organization.
Default Currency ISO Code	User’s default currency setting for new records. Available only for organizations that use multiple currencies.
Department	Group that user works for, for example, Customer Support. Up to 80 characters are allowed in this field.
Email	Email address of user. Must be a valid email address in the form: jsmith@acme.com. Up to 80 characters are allowed in this field.
Email Encoding	Character set and encoding for outbound email sent by user from within Database.com. ISO-8859-1 represents all Latin characters and should be used by English-speaking users. UTF-8 (Unicode) represents all characters for all of the world’s languages, but is not supported by some older email software. Shift_JIS, EUC-JP and ISO-2022-JP are useful for Japanese users.

Field	Description
Employee Number	Identifying number for a user.
End of day	Time of day that user generally stops working. Used to define the times that display in the user's calendar.
Fax	Fax number for user.
Federation ID	The value used to identify a user for federated authentication single sign-on. For more information, see Configuring SAML Settings for Single Sign-On on page 278. Also used with identity providers. See About Identity Providers and Service Providers on page 300
First Name	First name of user, as displayed on the user edit page. Up to 40 characters are allowed in this field.
Force.com Quick Access Menu	Enables the quick access menu, which appears in object list view pages, and provides shortcuts to customization features.
Language	The primary language for the user. All text and online help is displayed in this language. The language and locale used to display values such as numbers and date formats in the application's user interface.
Last Login	Date of last login. (Read only)
Last Name	Last name of user, as displayed on the user edit page. Up to 80 characters are allowed in this field.
Locale	Country or geographic region in which user is located.
	The <code>Locale</code> setting affects the format of date, date/time, and number fields. For example, dates in the English (United States) locale display as 06/30/2000 and as 30/06/2000 in the English (United Kingdom) locale. Times in the English (United States) locale display using a twelve-hour clock with AM and PM (for example, 2:00 PM), whereas in the English (United Kingdom) locale, they display using a twenty-four-hour clock (for example, 14:00). The <code>Locale</code> setting also affects the first and last name order on Name fields for users. For example, Bob Johnson in the English (United States) locale displays as Bob Johnson, whereas the Chinese (China) locale displays the name as Johnson Bob.
Mobile	Cellular or mobile phone number. Up to 40 characters are allowed in this field.
Modified By	User who last changed the user fields, including modification date and time. (Read only)
Name	Combined first and last name of user, as displayed on the user detail page.

Field	Description
Newsletter	Opt in to receive user-targeted promotional emails from salesforce.com. This field is not available if your organization has disabled your choice to receive emails from salesforce.com.
Phone	Phone number of user. Up to 40 characters are allowed in this field.
Profile	Administrative field that defines a user's permission to perform different functions within the application.
Role	Administrative field that specifies position of user within an organization, for example, Western Region Support Manager. Roles are selected from a picklist of available roles, which can be changed by an administrator.
Send Apex Warning Emails	If selected, specifies that the user will receive email notification whenever he or she executes Apex that surpasses more than 50% of allocated governor limits. See the Database.com Apex Code Developer's Guide for information.
Start of day	Time of day that user generally starts working. Used to define the times that display in the user's calendar.
State/Province	State or province portion of user's address. Up to 20 characters are allowed in this field.
Time Zone	Primary time zone in which user works. Users in Arizona should select the setting with "America/Phoenix," and users in parts of Indiana that do not follow Daylight Savings Time should select the setting with "America/Indianapolis."
Title	Job title of user. Up to 80 characters are allowed in this field.
Used Space	Amount of disk storage space the user is using. See Monitoring Resources on page 62.
User License	Indicates the type of user license. For more information about user licenses, see Understanding User License Types on page 197.
Username	Administrative field that defines the user's login. Up to 80 characters are allowed in this field.
Zip/Postal Code	Zip code or postal code portion of user's address. Up to 20 characters are allowed in this field.

Managing Your Security

Changing Your Password

To change your password at any time, click **My Personal Information > Change My Password**.



Note: If you have the “User Single Sign-On” permission, only an administrator can reset your password. Please contact your administrator for assistance. For information about Single Sign-On, see [About Single Sign-On](#) on page 289.

When you change your password, if you have not previously selected and answered a security question, you are prompted to do so. You must answer this question correctly if you ever forget your password and need it to be reset.

Additional Password Considerations

As you enter a new password in the **New Password** field, a visual indicator provides dynamic feedback on the strength of that password. When the password matches the minimum requirements for your organization's password policy, the visual indicator and associated text indicate that the password is acceptable and can now be saved. A tip is displayed to suggest how to make the password stronger and more difficult to guess.

You might have to activate your computer to successfully log in to Database.com whenever your password is changed or reset, or when you log in from a computer you have not used to access Database.com before. Activating your computer allows Database.com to verify your identity and prevent unauthorized access. To activate your computer:

1. When prompted on the login page, click the **Send Activation Link** button. Database.com sends an activation email to the email address specified on your Database.com user record.
2. When you receive the activation email, copy and paste the activation link into your browser.

The activation link included in the email is available for you to copy and paste into your browser up to 24 hours from the time you clicked the **Send Activation Link** button. After 24 hours, the activation link expires, and you must repeat the activation process to log in.

Activating Your Computer

You might have to activate your computer to successfully log in to Database.com whenever your password is changed or reset, or when you log in from a computer you have not used to access Database.com before. Activating your computer allows Database.com to verify your identity and prevent unauthorized access. To activate your computer:

1. When prompted on the login page, click the **Send Activation Link** button. Database.com sends an activation email to the email address specified on your Database.com user record.
2. When you receive the activation email, copy and paste the activation link into your browser.

The activation link included in the email is available for you to copy and paste into your browser up to 24 hours from the time you clicked the **Send Activation Link** button. After 24 hours, the activation link expires, and you must repeat the activation process to log in.

Retrieving Forgotten Passwords

Follow these steps if you have forgotten your password:

1. Go to <https://login.database.com>.
2. Click **Forgot your password?**.
3. Enter your username and click **Continue**. A message is automatically sent to your email address.
4. Click the link provided in that message, answer your password question, and click **Continue**.
5. A temporary password is automatically sent to your email address. Click the link to log in using that temporary password.
6. When prompted, enter a new password.

You might have to activate your computer to successfully log in to Database.com whenever your password is changed or reset, or when you log in from a computer you have not used to access Database.com before. Activating your computer allows Database.com to verify your identity and prevent unauthorized access. To activate your computer:

1. When prompted on the login page, click the **Send Activation Link** button. Database.com sends an activation email to the email address specified on your Database.com user record.
2. When you receive the activation email, copy and paste the activation link into your browser.

The activation link included in the email is available for you to copy and paste into your browser up to 24 hours from the time you clicked the **Send Activation Link** button. After 24 hours, the activation link expires, and you must repeat the activation process to log in.

Resetting Your Security Token

A security token is an automatically generated key that you must add to the end of your password in order to log in to Database.com from an untrusted network. For example, if your password is mypassword, and your security token is XXXXXXXXXXXX, then you must enter mypasswordXXXXXXXXXX to log in. Security tokens are required whether you log in via the API or the Data Loader.

You are offered a security token if you try to access Database.com from an untrusted network. Once you have been issued a security token, you have the option to reset this security token at any time.

To reset your security token, click **My Personal Information > Reset Security Token**, and click the **Reset My Security Token** button. The new security token is sent via email to the email address on your Database.com user record.

If you have never been offered a security token, for example, because your organization restricts the IP addresses from which you can log in, the **Reset My Security Token** node does not appear under **My Personal Information**.

 **Tip:** We recommend that you obtain your security token using the Database.com user interface from a trusted network prior to attempting to access Database.com from a new IP address.

Granting Login Access

To assist you, your administrator or a customer support representative may need to log in to the application using your login. You can grant access to them for a specified duration. For security reasons, the maximum period for granting access is limited to one year. During the time you have granted access, they can use your login and access your data to help you resolve any problems.

To grant login access:

1. Choose **My Personal Information > Grant Login Access**.
2. Set the access expiration date by choosing a value from the picklist.
3. Click **Save**.

If an administrator or support representative makes setup changes using your login, the setup audit trail lists those changes, including the username of the delegate user who made the changes.

 **Note:** You may be unable to grant access to certain support organizations due to restrictions set up by your administrator.

Managing Custom Views

Creating Custom List Views

User Permissions Needed	
To create custom list views:	"Read" on the type of record included in the list

You can create new list views to see a specific set of records in your organization, such as users, Apex classes, and workflow rules.

To edit or delete any view you created, click **Edit** next to the View drop-down list.

To create a new view, click **Create New View** at the top of any list page. Fill in the following:

1. Enter View Name

Enter the name to appear in the View drop-down list.

2. If you have the "Customize Application" permission, enter a unique name to be used by the API.

3. Specify Filter Criteria

The available filtering criteria differs depending on the records you're filtering.

4. Select Fields to Display

The default fields are automatically selected. You can choose up to 15 different fields to display in your view. When you select a long text area field, up to 255 characters are displayed in the list view.

a. To add or remove fields, select a field name, and click the **Add** or **Remove** arrow.

b. Use the arrows to arrange the fields in the proper sequence.

5. Restrict Visibility

Specify whether all administrative users or just you can see the custom view. To see a list view, users must also have the appropriate "Read" permission on the type of records within the list view.

6. Click Save. The view appears in the View drop-down list so you can access it later.

You can rename an existing list view and click **Save As** to save the criteria of the list view without altering the original view.

To navigate back to the last list page you viewed, click **Back to list** at the top of any detail page.



Note: The information you see in list views is only the data to which you have access—either records you own or have read or read/write access to, records that have been shared to you, or records owned by or shared with users in roles below you in the role hierarchy.

In addition, you can view only those fields that are visible in your field-level security settings.

Deleting List Views

To delete one of your custom views, select the view from the drop-down list and click the **Edit** link. At the top of the page, click the **Delete** button.

Navigating Long Lists

Some pages in the Database.com Console include the following tools for managing a large amount of data:

- To show a filtered list of items, select a predefined list from the View drop-down list, or click **Create New View** to define your own [custom view](#).
- To edit or delete any view you created, select it from the View drop-down list and click **Edit**.
- At the top of a list, click a letter to show items that correspond to that letter, or click **Other** to show items whose names begin with numbers or symbols. Click **All** to display all items that match the criteria of the current view.
- To sort list view items by the data in a particular column, click that column's heading. This sorts text data alphabetically and numerical data in ascending order. Dates are sorted with the most recent date first. To reverse the sort order, click the column heading a second time.



Note: You can sort by any custom field except multi-select picklists.

- Click the **Previous Page** or **Next Page** link to go to the previous or next set of items in the current view.
- At the bottom of a list, click the **fewer** or **more** link to decrease or increase the number of items per page.

In some related lists with many items, the following links are available:

- Click **Show [number] more** to increase the number of items in the list.
- Click **Go to list** to display a secondary page of the entire related list.

Entering Filter Criteria

For each filter, set the field, operator, and value.

To enter filter criteria:

- Choose a field from the first drop-down list.
- Choose a filter operator.
- Enter a value in the third field and click **OK**.
- Optionally, click **Add > Filter Logic** to change the default AND relationship between each filter.

Filtering on Special Picklist Values

When creating filter criteria, you can use special picklist values for your search criteria. These are special picklists with values of either "True" or "False."



Note: If you're creating filter criteria for a list view, the lookup icon automatically displays when you choose to filter on one of the special picklists. Click the lookup icon to choose the value "True" or "False." Alternatively, you can manually enter "True" or "False" in the filter criteria.

These are the available fields and their values:

Special Picklist Field	Value	Description
Users: Active	True	User is active and can log in.

Special Picklist Field	Value	Description
	False	User is inactive and cannot log in.

 **Note:** The special picklists you can view are only those that are visible in your field-level security settings.

Special Date Values for Filter Criteria

Use the following special date values when defining filter criteria using dates. If you choose a date field (for example, Create Date, Last Update Date) as a filter criteria, the value can be a date in the format allowed by your Locale setting or any of the following special values:

Special Date Value	Range
YESTERDAY	Starts at 12:00:00 a.m. on the day before the current day and continues for 24 hours.
TODAY	Starts at 12:00:00 a.m. on the current day and continues for 24 hours.
TOMORROW	Starts at 12:00:00 a.m. on the day after the current day and continues for 24 hours.
LAST WEEK	Starts at 12:00:00 a.m. on the first day of the week before the current week and continues for seven days. The week is defined by the Locale drop-down list in your personal information settings at My Personal Information > Personal Information .
THIS WEEK	Starts at 12:00:00 a.m. on the first day of the current week and continues for seven days. The week is defined by the Locale drop-down list in your personal information settings at My Personal Information > Personal Information .
NEXT WEEK	Starts at 12:00:00 a.m. on the first day of the week after the current week and continues for seven days. The week is defined by the Locale drop-down list in your personal information settings at My Personal Information > Personal Information .
LAST MONTH	Starts at 12:00:00 a.m. on the first day of the month before the current month and continues for all the days of that month.
THIS MONTH	Starts at 12:00:00 a.m. on the first day of the current month and continues for all the days of that month.
NEXT MONTH	Starts at 12:00:00 a.m. on the first day of the month after the current month and continues for all the days of that month.
LAST 90 DAYS	Starts at 12:00:00 a.m. 90 days before the current day and continues up to the current second. (The range includes today.)

Special Date Value	Range
NEXT 90 DAYS	Starts at 12:00:00 a.m. on the day after the current day and continues for 90 days. (The range does not include today.)
LAST n DAYS	Starts at 12:00:00 a.m. n days before the current day and continues up to the current second. (The range includes today.)
NEXT n DAYS	Starts at 12:00:00 a.m. on the next day and continues for the next n days. (The range does not include today.)
LAST QUARTER	Starts at 12:00:00 a.m. on the first day of the quarter before the current quarter and continues to the end of that quarter.
THIS QUARTER	Starts at 12:00:00 a.m. on the first day of the current quarter and continues to the end of the quarter.
NEXT QUARTER	Starts at 12:00:00 a.m. on the first day of the quarter after the current quarter and continues to the end of that quarter.
LAST n QUARTERS	Starts at 12:00:00 a.m. on the first day of the quarter n quarters ago and continues to the end of the quarter before the current quarter. (The range does not include the current quarter.)
NEXT n QUARTERS	Starts at 12:00:00 a.m. on the first day of the quarter after the current quarter and continues to the end of the quarter n quarters in the future. (The range does not include the current quarter.)
LAST YEAR	Starts at 12:00:00 a.m. on January 1 of the year before the current year and continues through the end of December 31 of that year.
THIS YEAR	Starts at 12:00:00 a.m. on January 1 of the current year and continues through the end of December 31 of the current year.
NEXT YEAR	Starts at 12:00:00 a.m. on January 1 of the year after the current year and continues through the end of December 31 of that year.
LAST n YEARS	Starts at 12:00:00 a.m. on January 1 of the year n years ago and continues through December 31 of the year before the current year.
NEXT n YEARS	Starts at 12:00:00 a.m. on January 1 of the year after the current year and continues through the end of December 31 of the n th year.
LAST FISCAL QUARTER	Starts at 12:00:00 a.m. on the first day of the fiscal quarter before the current fiscal quarter and continues through the last day of that fiscal quarter. The fiscal quarter is defined in the company profile at Company Profile > Fiscal Year .
THIS FISCAL QUARTER	Starts at 12:00:00 a.m. on the first day of the current fiscal quarter and continues through the end of the last day of the current fiscal quarter. The fiscal quarter is defined in the company profile at Company Profile > Fiscal Year .
NEXT FISCAL QUARTER	Starts at 12:00:00 a.m. on the first day of the fiscal quarter after the current fiscal quarter and continues through the last day of that fiscal quarter. (The

Special Date Value	Range
	range does not include the current quarter.) The fiscal quarter is defined in the company profile at Company Profile > Fiscal Year .
LAST n FISCAL QUARTERS	Starts at 12:00:00 a.m. on the first day of the fiscal quarter n fiscal quarters ago and continues through the end of the last day of the previous fiscal quarter. (The range does not include the current fiscal quarter.) The fiscal quarter is defined in the company profile at Company Profile > Fiscal Year .
NEXT n FISCAL QUARTERS	Starts at 12:00:00 a.m. on the first day of the fiscal quarter after the current fiscal quarter and continues through the end of the last day of the n th fiscal quarter. (The range does not include the current fiscal quarter.) The fiscal quarter is defined in the company profile at Company Profile > Fiscal Year .
LAST FISCAL YEAR	Starts at 12:00:00 a.m. on the first day of the fiscal year before the current fiscal year and continues through the end of the last day of that fiscal year. The fiscal year is defined in the company profile at Company Profile > Fiscal Year .
THIS FISCAL YEAR	Starts at 12:00:00 a.m. on the first day of the current fiscal year and continues through the end of the last day of the fiscal year. The fiscal year is defined in the company profile at Company Profile > Fiscal Year .
NEXT FISCAL YEAR	Starts at 12:00:00 a.m. on the first day of the fiscal year after the current fiscal year and continues through the end of the last day of that fiscal year. The fiscal year is defined in the company profile at Company Profile > Fiscal Year .
LAST n FISCAL YEARS	Starts at 12:00:00 a.m. on the first day of the fiscal year n fiscal years ago and continues through the end of the last day of the fiscal year before the current fiscal year. (The range does not include the current fiscal year.) The fiscal year is defined in the company profile at Company Profile > Fiscal Year .
NEXT n FISCAL YEARS	Starts at 12:00:00 a.m. on the first day of the fiscal year after the current fiscal year and continues through the end of the last day of the n th fiscal year. (The range does not include the current fiscal year.) The fiscal year is defined in the company profile at Company Profile > Fiscal Year .

Browsing and Searching Setup

User Permissions Needed	
To view setup pages:	“View Setup and Configuration”

The sidebar includes a search box for browsing and quickly finding setup tools. You can:

- Type the first few characters of a setting name in the **Quick Find** box. As you type, items that match your search terms appear in the menu. Click an item in the list to go to its setup page.

For example, to quickly find the user profiles page, type prof in the **Quick Find** box.

- Click **Expand All** to open all setup menus.
If you have typed anything in the **Quick Find** box, only the menus with matching items are expanded.
- Click **Collapse All** to close all setup menus.

Managing the Database Schema

Managing Objects

Managing Custom Objects

Custom Objects Deployment Status

User Permissions Needed	
To deploy custom objects:	“Customize Application”

While developing custom objects, you may not want users to see and interact with a new object. Because users may get frustrated by losing data when you delete custom fields, control visibility of the new object until you are finished.

Use the `Deployment Status` setting in the custom object definition to control when users can see and use a custom object.

- Choose “In Development” as the `Deployment Status` when first creating your custom object to hide it from users while you are designing and testing it. Making the status “In Development” hides the custom objects from search results from all users, except those with the “Customize Application” permission.
- Change the `Deployment Status` to “Deployed” when you want to allow all users to use the custom object.
- After deploying a custom object, change the `Deployment Status` back to “In Development” if you want to make more enhancements to it.

Managing Custom Objects

User Permissions Needed	
To create and edit custom objects:	“Customize Application”

After creating your custom objects, you can customize, edit, and delete them. Click **Create > Objects** to display the Custom Objects list page, which shows the list of custom objects defined for your organization. From the Custom Objects list page, you can:

- Click **New Custom Object** to [define a custom object](#).
- Click the object name to display detailed information about the custom object and customize it further.
- To update the custom object definition, click **Edit** and update the desired fields in the wizard.
- To [delete a custom object](#), click **Del**.



Note: You can't delete more than 100,000 combined objects and child records at the same time. To delete an object that has more than 100,000 child records, first delete an appropriate number of its child records.

- To view deleted custom objects, click the **Deleted Objects** link. The total number of deleted custom objects for your organization is listed in parentheses.

When viewing the detail page of a custom object, the lower portion page provides information about various characteristics of the custom object: standard fields, custom fields, field history tracking, and relationships. You can:

- Click on individual items to display additional detail.
- Click **more** at the bottom of the page or **View More** below a related list to display more items.
- Click **New** to directly add new items.

Defining Custom Objects

User Permissions Needed	
To create and edit custom objects:	"Customize Application"

Define custom objects to track and store data unique to your organization. For the total number of custom objects you can create, see [Getting Started with Database.com](#).

To create a custom object:

- Click **Create > Objects**.
- Click **New Custom Object**, or click **Edit** to modify an existing custom object.
- Enter the following:

Field	Description
Label	A name used to refer to the object in any user interface pages.
Plural Label	The plural name of the object.
Gender	If it is appropriate for your organization's default language, specify the gender of the label. This field appears if the organization-wide default language expects gender. Your personal language preference setting does not affect whether the field appears. For example, if the organization's default language is English and your personal language is French, you are not prompted for gender when creating a custom object.
Starts with a vowel sound	If it is appropriate for your organization's default language, check if your label should be preceded by "an" instead of "a."
Object Name	A unique name used to refer to the object when using the API. The Object Name field can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
Description	An optional description of the object. A meaningful description will help you remember the differences between your custom objects when you are viewing them in a list.

Field	Description
Record Name	The name used in list views and related lists.
Data Type	The type of field (text or auto-number) for the record name. Records that have unique IDs instead of names use auto-numbers. An auto-number is a unique number assigned automatically. It is always a read-only field.
Display Format	For a Record Name of type auto-number, enter a display format. You can have up to two sets of curly braces. For more details about display format options, see Custom Field Attributes on page 102.
Starting Number	For a Record Name of type auto-number, enter the number to use when creating your first record for this custom object.
Track Field History	Enables your organization to track changes to fields on the custom object records, such as who changed the value of a field, when it was changed, and what the value of the field was before and after the edit. History data is available for reporting, so users can easily create audit trail reports when this feature is enabled.
Deployment Status	Indicates whether the custom object is visible to other users. For more information, see Custom Objects Deployment Status on page 85.

4. Click **Save**.

Notes on Custom Objects

Object Permissions

When you create a custom object, the “Read,” “Create,” “Edit,” and “Delete” permissions for that object are disabled for any profiles in which “View All Data” or “Modify All Data” is disabled. You can enable access to custom objects in permission sets or custom profiles, and assign them to the users who need access. For more information, see [Overview of User Permissions and Access](#) on page 207.

Sharing Model

The data sharing model for all custom objects is controlled by an organization-wide default setting. For more information, see [Custom Object Security](#) on page 90.

Delegating Custom Object Administration

After you create a custom object, you can [delegate the administration](#) of the custom object to other non-administrator users.

Queues

After you create a custom object, you can define [queues](#) to distribute ownership of custom object records to your users.

Defining Custom Object Fields

User Permissions Needed	
To view and edit standard fields:	“Customize Application”
To create custom fields:	“Customize Application”

Custom object fields store the data for your custom object records.

Custom Fields for Custom Objects

You can create custom fields to store information unique to your organization; see [Creating Custom Fields](#) on page 98. You can also create custom relationship fields to associate your custom object with another object in Database.com. For more information about relationships, see [Overview of Relationships](#) on page 91.

Standard Fields for Custom Objects

Custom objects automatically include the following standard fields. Click **Edit** to modify any of the editable fields.

Field	Description
Created By	User who created the record.
Currency	Currency of the record if multicurrency is enabled.
Last Modified By	User who most recently changed the record.
Name	Identifier for the custom object record. This name appears in related lists and lookup dialogs.
Owner	Assigned owner of the custom object record. If the custom object becomes the detail side of a master-detail relationship, this field is removed, as ownership of the data is controlled by the master object, or by the primary master object for a custom object with two master-detail relationships.
	 Note: Custom objects on the “detail” side of a master-detail relationship can't have sharing rules, manual sharing, or queues, as these require the Owner field.

Deleting Custom Objects

User Permissions Needed	
To delete custom objects:	“Customize Application”

To delete custom objects:

1. Click **Create > Objects**.
2. Next to the custom object, click **Del**.

- When prompted, select the Yes, I want to delete the custom object checkbox to confirm, and click **Delete**.

Notes on Deleting Custom Objects

- When you delete a custom object, Database.com does not add it to the Recycle Bin with other deleted records. Instead, deleted objects appear in the Deleted Custom Objects list for 45 days. During this time you can restore an object and the data stored in it, or permanently erase the object and its data. After 45 days, the object and its data are permanently erased. For more information on restoring custom objects, see [Managing Deleted Custom Objects](#) on page 89.
- Until permanently erased, the custom object and its data still count against the maximum number of items allowed in your organization. To view a list of the maximum number of custom objects, fields, and other items allowed in your organization, see [Getting Started with Database.com](#).
- When you delete a custom object, Database.com does the following:
 - If the deleted object is on the detail side of a master-detail relationship, changes the master-detail relationship to a lookup relationship.
 - Permanently erases:
 - The object's data records currently in the Recycle Bin
 - List views for the object
 - Workflow rules and actions that use the object
 - Hides, inactivates, or disables:
 - The custom object definition and all related definitions
 - The object's data records and all related data records
 - Custom formula fields on the object
 - Custom validation rules on the object

Database.com restores these items if you restore the custom object.

- You can't delete a custom object if it is on the master side of a master-detail relationship. When you delete a custom object that is on the detail side of a master-detail relationship, the relationship is converted to a lookup relationship. If you restore the custom object, you must manually convert it to a master-detail. See [Changing Custom Field Type](#) on page 113.
- You can't delete a custom object if it is referenced in an [Apex script](#).
- In a many-to-many relationship, a user can't delete a parent record if there are more than 200 junction object records associated with it *and* if the junction object has a roll-up summary field that rolls up to the other parent. To delete this object, manually delete junction object records until the count is fewer than 200.
- You can't delete more than 100,000 combined objects and child records at the same time. To delete an object that has more than 100,000 child records, first delete an appropriate number of its child records.

Managing Deleted Custom Objects

User Permissions Needed	
To restore deleted custom objects:	"Customize Application"
To permanently delete custom objects:	"Customize Application"

Until permanently erased, the custom object and its data still count against the maximum number of items allowed in your organization.

To view a list of your deleted custom objects:

- Click **Create > Objects**.

2. Click **Deleted Objects** at the bottom of the list of custom objects. The number in parentheses indicates the total number of deleted custom objects in your organization. This link only displays when you have a deleted custom object.
3. Use the list of deleted custom objects to perform the following actions:
 - To view details about an object, click the label.
 - To permanently remove the custom object and its data, click **Erase**.
 - To restore the object and its data, click **Undelete**. Some attributes of deleted custom objects are not automatically restored. To restore these manually, complete the following steps:
 - ◊ The developer name was changed to `objectname_del`. Edit the object to change the developer name.
 - ◊ The custom object's Deployment Status field was set to In Development. When all changes impacted by the delete have been restored, edit the object to change the status to Deployed.
 - ◊ Rebuild any workflow rules on the object.
 - ◊ Reactivate any custom validation rules for the object.
 - ◊ Open and save any custom formula fields on the custom object to enable them.
 - ◊ If the deleted custom object is on the detail side of a master-detail relationship:
 - Master-detail fields are converted to lookup fields when the object is deleted. Change the field types back to master-detail.

 **Note:** It may take several hours before you can search for records in the object.

Custom Object Security

Many different security settings work together so you can control access to your custom objects with great flexibility. Set custom object security at the following different levels:

- **Object**—set the access users have to create, read, edit, and delete records for each object.
- **Records**—set the default sharing model for all your users. This determines the access users have to custom object records that they do not own.
- **Relationship**—for objects on the detail side of a master-detail relationship, specify the sharing access that users must have to the master record in order to create, edit, or delete the associated detail records. This is specified in the `Sharing Setting` attribute of the master-detail relationship field on the detail object.
- **Fields**—set the level of access users have to fields.

The following requirements apply to custom objects with no master-detail relationship:

Action	Required Privileges
Create a record	“Create” permission.
View a record	“Read” permission and Public Read Only or Public Read/Write sharing model if not the record owner.
Edit a record	“Edit” permission and Public Read/Write sharing model if not the record owner.
Delete a record	“Delete” permission and must be the record owner or above the record owner in the role hierarchy.

The following requirements apply to custom objects that have a master-detail relationship with a custom object:

Action	Required Privileges
Create a record	“Create” permission and either read or read/write access to the related master record, depending on the value of the Sharing Setting attribute of the master-detail relationship field on the detail object.
View a record	“Read” permission and read access to the related master record. If the record has two master records in a many-to-many relationship, the user must have read access to both master records.
Edit a record	“Edit” permission and either read or read/write access to the related master record, depending on the value of the Sharing Setting attribute of the master-detail relationship field on the detail object.
Delete a record	“Delete” permission and either read or read/write access to the related master record, depending on the value of the Sharing Setting attribute of the master-detail relationship field on the detail object. When a user deletes a record that has related custom object records, all related custom object records are deleted regardless of whether the user has delete permission to the custom object.

[Delegated administrators](#) can manage nearly every aspect of specified custom objects, but they cannot create or modify relationships on the object or set organization-wide sharing defaults.

Managing Relationships

Overview of Relationships

Use relationships to associate an object with other objects in Database.com. For example, you can define a relationship between a Line Item custom object and the Invoice Statement custom object to which it belongs. You can define different types of relationships by creating custom relationship fields on an object. Before you begin [creating relationships](#), determine the type of relationship that suits your needs.

There are different types of relationships between objects in Database.com. Their differences include how they handle data deletion, record ownership, and security:

Master-detail

This type of relationship closely links objects together such that the master record controls certain behaviors of the detail and subdetail record. For example, you can define a two-object master-detail relationship, such as Merchandise—Invoice Statement, that extend the relationship to subdetail records, such as Merchandise—Invoice Statement—Line Item. You can then perform operations across the master—detail—subdetail relationship.

Behaviors of master-detail relationships include:

- When a master record is deleted, the related detail and subdetail records are also deleted.

- The `Owner` field on the detail and subdetail records is not available and is automatically set to the owner of the master record. Custom objects on the “detail” side of a master-detail relationship can't have sharing rules, manual sharing, or queues, as these require the `Owner` field.
- The security settings for the master record control the detail and subdetail records.
- As a best practice, don't exceed 10,000 child records for a master-detail relationship.



Tip: If you have a custom object called “Line Item” and you want each line item record deleted along with its associated invoice statement record, create a master-detail relationship on the Line Item custom object with Invoice Statement as the master object.

Many-to-many

You can use master-detail relationships to model *many-to-many* relationships between two objects. A many-to-many relationship allows each record of one object to be linked to multiple records from another object and vice versa. For example, you may have a bug object that relates to a support case object such that a bug could be related to multiple support cases and a support case could also be related to multiple bugs. See [Creating a Many-to-Many Relationship](#) on page 93.

Lookup

This type of relationship links two objects together, but it has no effect on record ownership or security.

Use lookup relationships to:

- Link two different objects.
- Link an object with itself (with the exception of the user object; see [Hierarchical](#) on page 92). For example, you may want to link a custom object called “Bug” with itself to show how two different bugs are related to the same problem.

You can't delete an object or record in a lookup relationship if the combined number of records between the two linked objects is more than. To delete an object or record in a lookup relationship, first delete an appropriate number of its child records.

Hierarchical

This type of relationship is a special lookup relationship available only for the user object. It allows users to use a lookup field to associate one user with another that does not directly or indirectly refer to itself. For example, you can create a custom hierarchical relationship field to store each user's direct manager.

About Dependent Lookups

User Permissions Needed	
To manage dependent lookups:	“Customize Application”

A *dependent lookup* is a relationship field with a lookup filter that references fields on the source object.

When a user changes the value of a referenced field on the source object, Database.com immediately verifies that the value in the dependent lookup still meets the lookup filter criteria. If the value doesn't meet the criteria, an error message is displayed and users can't save the record until the value is valid.

If the referenced field on the source object is a lookup, master-detail, or hierarchy field, users can't change its value by typing. Instead, users must click the lookup icon and select a value in the lookup search dialog.

Creating a Many-to-Many Relationship

User Permissions Needed	
To create a many-to-many relationship:	"Customize Application"

You can use master-detail relationships to model *many-to-many* relationships between two objects. A many-to-many relationship allows each record of one object to be linked to multiple records from another object and vice versa. For example, you may have a bug object that relates to a support case object such that a bug could be related to multiple support cases and a support case could also be related to multiple bugs. When modeling a many-to-many relationship, you use a *junction object* to connect the two objects you want to relate to each other.

Junction Object

A custom object with two master-detail relationships. Using a custom junction object, you can model a “many-to-many” relationship between two objects. For example, you may have a custom object called “Bug” that relates to the standard Case object such that a bug could be related to multiple cases and a case could also be related to multiple bugs.

Creating the many-to-many relationship consists of:

1. [Creating the junction object.](#)
2. [Creating the two master-detail relationships.](#)

Creating the Junction Object

To create the junction object:

1. Click **Create > Objects**.
2. Click **New Custom Object**.
3. In the custom object wizard, consider these tips specifically for junction objects:
 - Name the object with a label that indicates its purpose, such as BugSupportCaseAssociation.
 - For the Record Name field, it is recommended that you use the auto-number data type.

Creating the Two Master-Detail Relationships

To create the two master-detail relationships:

1. Verify that the two objects you want to relate to each other already exist. For example, you may want to relate a support case object to a bug object.
2. On the junction object, create the first master-detail relationship field. In the custom field wizard:
 - a. Choose **Master-Detail Relationship** as the field type.
 - b. Select one of the objects to relate to your junction object. For example, select **SupportCase**.

The first master-detail relationship you create on your junction object becomes the *primary* relationship. Therefore, the junction object records inherit the value of the **Owner** field from their associated primary master record. Because objects on the detail side of a relationship do not have a visible **Owner** field, this is only relevant if you later delete both master-detail relationships on your junction object.
- c. Select a **Sharing Setting** option. For master-detail relationship fields, the **Sharing Setting** attribute determines the sharing access that users must have to a master record in order to create, edit, or delete its associated detail records.
3. On the junction object, create the second master-detail relationship. In the custom field wizard:
 - a. Choose **Master-Detail Relationship** as the field type.

- b. Select the other desired master object to relate to your junction object. For example, select Bug.

The second master-detail relationship you create on your junction object becomes the *secondary* relationship. If you delete the primary master-detail relationship or convert it to a lookup relationship, the secondary master object becomes primary.

- c. Select a *Sharing Setting* option. For master-detail relationship fields, the *Sharing Setting* attribute determines the sharing access that users must have to a master record in order to create, edit, or delete its associated detail records.

Considerations for Relationships

Review the following considerations before creating relationships between objects:

Relationship Limits

Each custom object can have up to two master-detail relationships and many lookup relationships. Each relationship is included in the maximum number of custom fields allowed. For the total number of custom fields you can create, see [Getting Started with Database.com](#).

Changing and Converting Relationships

After you have created a relationship, you can't change which objects are related via that relationship. If you need to do this, delete the relationship and create a new relationship.

You can convert a master-detail relationship to a lookup relationship as long as no roll-up summary fields exist on the master object.

You can convert a lookup relationship to a master-detail relationship, but only if the lookup field in all records contains a value.

Self Relationships

You can create a relationship from an object to itself, but it must be a lookup relationship, and a single record can't be linked to itself. However, a record can indirectly relate to itself. For example, the Gift Ideas merchandise category can have the Electronic merchandise category selected in the lookup relationship, and the Electronic merchandise category can have the Gift Ideas merchandise category selected in the lookup relationship.

You can't create a many-to-many self relationship, that is, the two master-detail relationships on the junction object can't have the same master object.

Master-Detail Relationships

To create multilevel master-detail relationships, you need the "Customize Application" user permission.

When you define a master-detail relationship, the custom object on which you are working is the "detail" side.

If a custom object is detail or subdetail component in a master-detail relationship, it can't also be the master of a different master-detail relationship.

You can have up to three custom detail levels.

An object can appear once in multilevel master-detail relationships. For example, a subdetail object in one multilevel master-detail relationship can't also be the owner of the master object in another multilevel master-detail relationship. A subdetail object can't also be the master object of the subdetail object's detail object.

You can't create a master-detail relationship if the custom object already contains data. You can, however, create the relationship as a lookup and then convert it to master-detail if the lookup field in all records contains a value.

Converting relationships from lookup to master-detail, or from master-detail to lookup behaves the same as for two-object master-detail relationships. That is, the two linked objects in the detail-subdetail1, or subdetail1-subdetail2 relationship have the same conversion limits as the master-detail relationship.

Roll-up summary fields work as in two-object master-detail relationships. A master can roll up fields on detail records; however, it can't directly roll up fields on subdetail records. To achieve this, the detail record must have a roll-up summary field for the field on the subdetail record, allowing the master to roll up from the detail's roll-up summary field.

[Custom junction objects](#) can't have detail objects. That is, a custom junction object can't become the master object in a multilevel master-detail relationship.

Undeleting the master record also undeletes detail and subdetail records.

As a best practice, don't exceed 10,000 child records for a master-detail relationship.

Many-to-Many Relationships

Junction object records are deleted when either associated master record is deleted and placed in the Recycle Bin. If both associated master records are deleted, the junction object record is deleted permanently and can't be restored.

Sharing access to a junction object record is determined by a user's sharing access to both associated master records and the [Sharing Setting](#) option on the relationship field. See [Custom Object Security](#) on page 90. For example, if the sharing setting on both parents is Read/Write, then the user must have Read/Write access to *both* parents in order to have Read/Write access to the junction object. If, on the other hand, the sharing setting on both masters is Read-Only, a user with Read-Only rights on the master records would have Read/Write access to the junction object.

You can create workflow rules on junction objects; but you can't create outbound messages on junction objects.

In a many-to-many relationship, a user can't delete a parent record if there are more than 200 junction object records associated with it *and* if the junction object has a roll-up summary field that rolls up to the other parent. To delete this object, manually delete junction object records until the count is fewer than 200.

The first master-detail relationship you create on your junction object becomes the *primary* relationship. Therefore, the junction object records inherit the value of the `Owner` field from their associated primary master record. Because objects on the detail side of a relationship do not have a visible `Owner` field, this is only relevant if you later delete both master-detail relationships on your junction object.

The second master-detail relationship you create on your junction object becomes the *secondary* relationship. If you delete the primary master-detail relationship or convert it to a lookup relationship, the secondary master object becomes primary.

Roll-up summary fields that summarize data from the junction object can be created on both master objects.

Formula fields and validation rules on the junction object can reference fields on both master objects.

You can define Apex triggers on both master objects and the junction object.

A junction object can't be on the master side of another master-detail relationship.

You can't create a many-to-many self relationship, that is, the two master-detail relationships on the junction object can't have the same master object.

Transferring Custom Objects Between Users

Transferring Records Overview

User Permissions Needed	
To transfer multiple custom objects:	“Transfer Record”
	AND
	“Edit” on the object type

A record owner, or any user above the owner in the role hierarchy, can transfer a single record to another user. The following table describes the ways in which single and multiple records can be transferred to another user:

Method	Available for
Transfer multiple records by selecting the records from a list view and clicking Change Owner	Custom objects, which can belong either to a user or a queue
Transfer multiple records using the Mass Transfer tool	Custom objects

Ability to Change Ownership

- Users with the “Modify All Data” permission, or users with the “Modify All” permission for the given object, can transfer any record, regardless of who owns the record.
- To transfer a single record or multiple records from a list view, the new owner must have at least the “Read” permission on the object type. This rule does not apply if you use the mass transfer tool.

Mass Transferring Records

User Permissions Needed	
To mass transfer custom objects:	“Transfer Record”
	AND
	“Edit” on the object type

Use the Mass Transfer tool to transfer multiple custom objects from one user to another.



Note: To transfer any records that you do not own, you must have the required user permissions as well as read sharing access on the records.

- Choose **Data Management > Mass Transfer Records**.
- Click the link for the type of record to transfer.
- Optionally, fill in the name of the existing record owner in the `Transfer from` field.
- In the `Transfer to` field, fill in the name of new record owner.
- [Enter search criteria](#) that the records you are transferring must match.
- Click **Find**.
- Select the checkbox next to the records you want to transfer. Optionally, check the box in the column header to select all currently displayed items.



Note: If duplicate records are found, you must select only one of the records to transfer. Transferring duplicate records results in an error.

8. Click **Transfer**.

Managing Fields

Managing Custom Fields

Defining Custom Object Fields

User Permissions Needed	
To view and edit standard fields:	“Customize Application”
To create custom fields:	“Customize Application”

Custom object fields store the data for your custom object records.

Custom Fields for Custom Objects

You can create custom fields to store information unique to your organization; see [Creating Custom Fields](#) on page 98. You can also create custom relationship fields to associate your custom object with another object in Database.com. For more information about relationships, see [Overview of Relationships](#) on page 91.

Standard Fields for Custom Objects

Custom objects automatically include the following standard fields. Click **Edit** to modify any of the editable fields.

Field	Description
Created By	User who created the record.
Currency	Currency of the record if multicurrency is enabled.
Last Modified By	User who most recently changed the record.
Name	Identifier for the custom object record. This name appears in related lists and lookup dialogs.
Owner	Assigned owner of the custom object record. If the custom object becomes the detail side of a master-detail relationship, this field is removed, as ownership of the data is controlled by the master object, or by the primary master object for a custom object with two master-detail relationships.

Note: Custom objects on the “detail” side of a master-detail relationship can't have sharing rules, manual sharing, or queues, as these require the **Owner** field.

Defining Default Field Values

User Permissions Needed	
To view default field values:	"View Setup and Configuration"
To define or change default field values:	"Customize Application"

To define a default field value:

1. Begin by creating a custom field; see [Creating Custom Fields](#) on page 98. You can also define a default value for an existing custom field; see [Editing Fields](#) on page 105.
2. Choose the type of field to create and click **Next**. For a list of the types available for default values, see [About Default Field Values](#) on page 115.
3. Enter the attributes for the field.
4. Enter a default value or define a formula to calculate the default value:
 - a. Click **Show Formula Editor** to view the formula editor.
 - b. If you are building a formula in the **Advanced Formula** tab or for workflow or validation rules, click **Insert Field**, choose a field, and click **Insert**.

To create a basic formula that passes specific Database.com data, select the **Simple Formula** tab, choose the field type in the **Select Field Type** drop-down list, and choose one of the fields listed in the **Insert Field** drop-down list.



Tip: Build *cross-object formulas* to span to related objects and reference merge fields on those objects.

- c. To insert an operator, choose the appropriate operator icon from the **Insert Operator** drop-down list. Use the examples in [Operators and Functions](#) on page 132.
- d. Double-click a function to insert it in your formula. For a description of each operator and function, see [Operators and Functions](#) on page 132.
- e. Click **Check Syntax** to check your formula for errors.



Note: You can define a formula for default values only where appropriate. For example, the default value options for picklist and checkbox fields are limited to the options available for those types of fields, such as **Checked**, **Unchecked**, or **Use first value as default value**.

5. Click **Next**.
6. Set the field-level security to determine whether the field should be visible for specific profiles, and click **Next**.
7. Click **Save** to finish or **Save & New** to create more custom fields.



Note:

[Default values](#) should not be assigned to fields that are both [required](#) and [unique](#), as uniqueness errors may result.

Creating Custom Fields

User Permissions Needed	
To create or change custom fields:	“Customize Application”

Create custom fields to store the information that is important to your organization. Before you begin, determine the type of custom field you want to create. You can create many different [custom field types](#), including [lookup](#), [master-detail](#), and [hierarchical](#) relationships.

To add a custom field:

1. Navigate to the appropriate object:

- Click **Create > Objects**, and click the name of the custom object in the list.

2. Click **New** in the Custom Fields & Relationships section of the page.



Tip: From this section, you can also set [field dependencies](#) and [field history tracking](#) on custom objects.

3. Choose the [type of field](#) to create, and click **Next**.



Note:

- Some data types are only available for certain configurations. For example, the [Master-Detail Relationship](#) option is available only for custom objects when the custom object does not already have a master-detail relationship. Also, custom settings only allow a subset of the available data types.
- Relationship fields count towards custom field limits.
- The [Roll-Up Summary](#) option is only available on certain objects.
- Field types correspond to API data types. For more information, see “[API Data Types and API Field Types](#)” in the [Web Services API Developer’s Guide](#).

4. For [relationship fields](#), choose the object that you want to associate with it.

5. 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. Use the field name for merge fields when referencing the field from the API. For more information, see [Guidelines for Using Merge Fields](#) on page 109.



Tip:

6. Enter any [field attributes](#).

7. For [relationship fields](#), optionally create a lookup filter to limit the valid values and lookup dialog results for the field.

8. Click **Next** to continue.

9. Specify the field’s access settings for each profile, and click **Next**.

Access Level	Enabled Settings
Users can read and edit the field.	Visible

Access Level	Enabled Settings
Users can read but not edit the field.	Visible and Read-Only
Users can't read or edit the field.	None

**Note:**

- When you create a custom field, by default the field isn't visible or editable for portal profiles, unless the field is **universally required**.
- Profiles with “View Encrypted Data” permission are indicated with an asterisk.

10. Click **Save** to finish or **Save & New** to create more custom fields.



Note: Creating fields may require changing a large number of records at once.

Considerations for Universally Required Fields

Review the following considerations before making your custom fields universally required.

- For a list of the types of custom fields that you can make universally required, see [About Universally Required Fields](#) on page 193.
- Edit pages always display universally required fields, regardless of field-level security.
- Default values** should not be assigned to fields that are both **required** and **unique**, as uniqueness errors may result.
- You cannot make a field universally required if it is used by a field update that sets the field to a blank value. For details, see [Defining Field Updates](#) on page 416.
- Required fields may be blank on records that existed before making the field required. When a user updates a record with a blank required field, the user must enter a value in the required field before saving the record.

About Encrypted Custom Fields

Encrypted custom fields are text fields that can contain letters, numbers, or symbols but are encrypted. The value of an encrypted field is only visible to users that have the “View Encrypted Data” permission.

Before you begin working with encrypted custom fields, review the following implementation notes and best practices:

Implementation Notes

- 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 salesforce.com. See also [Managing Database.com Master Encryption Keys](#) on page 298.
- Encrypted custom fields cannot be unique, an external ID, or have default values.
- While other text fields can contain up to 255 characters, encrypted text fields are limited to 175 characters due to the encryption algorithm.
- Encrypted fields are not available for use in filters such as list views, roll-up summary fields, and rule filters.
- If you have created encrypted custom fields, make sure your organization has secure connections using SSL (Secure Sockets Layer) enabled. To enable this setting for your organization, see [Setting Session Security](#) on page 240.
- If you have the “View Encrypted Data” permission and you [grant login access](#) to another user, be aware that the other user will be able to see encrypted fields unmasked (in plain text).
- Only users with the “View Encrypted Data” permission can clone the value of an encrypted field when cloning that record.

Best Practices

- Encrypted fields are editable regardless of whether the user has the “View Encrypted Data” permission. Use [validation rules](#) or [field-level security settings](#) to prevent users from editing encrypted fields.
- You can still validate the values of encrypted fields using [validation rules](#) or Apex. Both work regardless of whether the user has the “View Encrypted Data” permission. Data for encrypted fields in the debug log is masked.
- Existing custom fields cannot be converted into encrypted fields nor can encrypted fields be converted into another data type. To encrypt the values of an existing (unencrypted) field, export the data, [create an encrypted custom field](#) to store that data, and import that data into the new encrypted field.
- Mask Type is not an input mask that ensures the data matches the Mask Type. Use [validation rules](#) to ensure that the data entered matches the mask type selected.
- Use encrypted custom fields only when government regulations require it because they involve additional processing and have search-related limitations.

About Dependent Lookups

User Permissions Needed	
To manage dependent lookups:	“Customize Application”

A *dependent lookup* is a relationship field with a lookup filter that references fields on the source object.

When a user changes the value of a referenced field on the source object, Database.com immediately verifies that the value in the dependent lookup still meets the lookup filter criteria. If the value doesn't meet the criteria, an error message is displayed and users can't save the record until the value is valid.

If the referenced field on the source object is a lookup, master-detail, or hierarchy field, users can't change its value by typing. Instead, users must click the lookup icon and select a value in the lookup search dialog.

Custom Field Attributes

Here is a description of the attributes (in alphabetical order) that make up a custom field entry:

Field	Description
# Visible Lines	For long text area fields, set the number of lines to be displayed on edit pages. You can display between 2 and 50 lines (the default is 6 lines). If the text does not fit in the specified number of visible lines, scroll bars will appear. Long text area fields are displayed in their entirety on detail pages.
Calculation Options	Option that determines how a roll-up summary field is recalculated after its properties change. Choose Automatic calculation to recalculate a field the next time it is displayed. Choose Force a mass recalculation of this field as a safety net option to force recalculation of the roll-up summary field values.
Child Relationship Name	The name used in API SOQL relationship queries.
Data Type	The data type of a field determines what type of information is in the field. For example, an field with the Number data type contains a positive or negative integer. For more information on data types, see Custom Field Types on page 110.

Field	Description
Decimal Places	For number, currency, and percent fields, the number of digits you can enter to the right of a decimal point, for example, 4.98 for an entry of 2. Note that the system rounds the decimal numbers you enter, if necessary. For example, if you enter 4.986 in a field with Decimal Places of 2, the number rounds to 4.99.
Default Value	The value to apply when a user creates a new record. For checkbox custom fields, choose Checked or Unchecked as the default value to indicate the default when creating new records. Default values should not be assigned to fields that are both required and unique , as uniqueness errors may result. See About Default Field Values on page 115.
Description	Text that describes the custom field. This description is for administration purposes only and does not display to users on record detail and edit pages that include the field.
Display Format	<p>For auto-number fields, enter a Display Format to control such formatting details as the minimum number of leading zeros as well as any prefix or suffix for the number.</p> <p>Begin by entering the required minimum: {0}. This is a placeholder for the auto-number without any leading zeros. Add any prefix to your number before this placeholder and insert any suffix text after the placeholder. Insert any date prefixes or suffixes in the form of {YY}, {YYYY}, {MM}, or {DD}, which always represent the create date of the record.</p> <p>For information on using auto-number formats when entering your Display Format, see Auto-Number Formatting Examples on page 105.</p>
External ID	<p>For each object that can have custom fields, you can set up to three custom text, number, or email fields as external IDs. An external ID field contains record identifiers from a system outside of Database.com.</p> <p>You can use an external ID field to update or upsert records using the API. When using the API, you can use this field to prevent duplicates by also marking the field as Unique.</p>
	<p>Note: Custom fields marked as Unique count against an object's limit of three External ID fields.</p>
Filter Criteria	The criteria used to select a group of records to calculate the value of a roll-up summary field .
Formulas	Enter the formula for the custom formula field. For help on building formulas, see Building Formulas on page 117.
Help Text	The text that displays in the field-level help hover text for this field.
Label	Name of the custom field as you want it to appear.
Length (for text fields)	For text fields, the maximum number of characters that a user can enter in a field (up to 255 characters).

Field	Description
Length (for number, currency, percent fields)	For number, currency, and percent fields, the number of digits you can enter to the left of the decimal point, for example, 123.98 for an entry of 3.
Mask Character	For encrypted text fields, determines the character to use for hidden characters. Available options are * and X.
Mask Type	For encrypted text fields, determines which characters are hidden and the use of dashes in the field. Masked characters are hidden using the character selected in Mask Character . Available options are:
	Mask All Characters
	All characters in the field are hidden.
	Last Four Characters Clear
	All characters are hidden but the last four display.
	Credit Card Number
	The first 12 characters are hidden and the last four display. Database.com automatically inserts a dash after every fourth character.
	National Insurance Number
	All characters are hidden. Database.com automatically inserts spaces after each pair of characters if the field contains nine characters. Use this option for UK NINO fields.
	Social Security Number
	The first five characters are hidden and the last four display. Database.com automatically inserts a dash after the third and fifth characters.
	Social Insurance Number
	All characters are hidden but the last three display. Database.com automatically inserts a dash after the third and sixth characters.
Master Object	The object on the master side of a master-detail relationship used to display the value of a roll-up summary field .
Related List Label	For relationship fields, the title for the related list that displays associated records on the parent record.
Related To	For relationship fields, the name of the associated object.
Required	Makes the field required everywhere in Database.com.
	Default values should not be assigned to fields that are both required and unique , as uniqueness errors may result. See About Universally Required Fields on page 193.

Field	Description	
Roll-Up Type	For roll-up summary fields , choose the type of calculation to make:	
Type	Description	
COUNT	Totals the number of related records.	
SUM	Totals the values in the field you select in the Field to Aggregate option. Only number, currency, and percent fields are available.	
MIN	Displays the lowest value of the field you select in the Field to Aggregate option for all directly-related records. Only number, currency, percent, date, and date/time fields are available.	
MAX	Displays the highest value of the field you select in the Field to Aggregate option for all directly-related records. Only number, currency, percent, date, and date/time fields are available.	
Starting Number	<p>For auto-number fields, enter a Starting Number that is less than 1 billion. Check Generate Auto Number for existing records... if you would like to automatically number all current records beginning with the starting number you enter. If unchecked, the next record you enter will be assigned the starting number and your existing records will be blank in this field.</p> <p>An auto-number field can contain up to 10 digits and up to 20 additional characters for your prefix or suffix.</p>	
Sharing Setting	For master-detail relationship fields, the Sharing Setting attribute determines the sharing access that users must have to a master record in order to create, edit, or delete its associated detail records.	
Summarized Object	The object on the detail side of a master-detail relationship used to provide the values calculated in a roll-up summary field .	
Unique	<p>If checked, prevents duplicate field values.</p> <p>For text fields, you can control whether values that are identical except for their case are considered unique. Select Treat "ABC" and "abc" as duplicate values to enforce case-insensitive uniqueness, or select Treat "ABC" and "abc" as different values to enforce case-sensitive uniqueness.</p>  <p>Note: Custom fields marked as Unique count against an object's limit of three External ID fields.</p>	

Field	Description
Values	For picklist fields, a list of available values (up to 255 characters for each value). For picklists, select the appropriate checkbox to alphabetize the picklist entries. You can also set the first value as the default selection. If you mark both boxes, Database.com first alphabetizes the entries and then sets the first alphabetized value as the default. For multi-select picklists, enter a list of values, check the sorting options that apply, and enter how many values you want displayed at a time on edit pages, which determines the box height.

Auto-Number Formatting Examples

Use these examples when setting the display format for auto-number fields.

Format	Displayed Values
{0}	3 66 103
{000}	003 066 103
Sample- {00000}	Sample- 00003 Sample- 00666 Sample- 10023
Value- {00} {MM} {DD} {YY}	Value- 03 12 02 04 Value- 76 03 03 04 Value- 123 11 09 04
PO #{0} {MM}-{DD}-{YY}	PO #12233 12-20-04 PO #25 06-07-04 PO #3 07-07-04

Editing Fields

User Permissions Needed
To create or change fields: “Customize Application”

1. Click **Create > Objects**, and click the name of the custom object in the list.
2. Click **Edit** next to the field's name.
3. Modify the [field attributes](#). The attributes differ depending on the field type.

To change the type of your custom field, see [Changing Custom Field Type](#) on page 113.

To make changes to picklists:

- Click **Edit** next to a value to change the name or make it the default picklist value.
- Click **Del** next to a value to remove it from the picklist.
- Click **New** to add values to the picklist. If you use record types, select any record types that you want to include the new values.
- Click **Reorder** to [change the sequence of picklist values](#).
- Click **Replace** to [change the values of picklist fields in existing records](#).
- Click **Printable View** to open an easy-to-print list of your picklist values.

4. Optionally, define custom help text for the field.
5. For lookup and master-detail relationship fields, optionally define a lookup filter.

6. For formula fields, click **Next** to modify the formula. See [Building Formulas](#) on page 117.
7. Click **Save**.

**Note:**

- Editing fields may require changing a large number of records at once.
- You cannot change the `Field Name` if a custom field is referenced in an [Apex](#) script.

Deleting Fields

User Permissions Needed	
To delete custom fields:	“Customize Application”

To delete a custom field:

1. Click **Create > Objects**, and click the name of the custom object in the list.
2. Click **Del** next to the name of the field.
3. When prompted, select the `Yes, I want to delete the custom field` checkbox to confirm, and click **Delete**.

Deleted custom fields and their data are stored until your organization permanently deletes them or 45 days has elapsed, whichever happens first. Until that time, you can restore the field and its data. For information on restoring deleted custom fields and relationships, see [Managing Deleted Custom Fields](#) on page 106.

**Note:**

- Before deleting a custom field, consider where it is referenced. You can't delete a custom field that is referenced elsewhere. For example, you cannot delete a custom field that is referenced by a field update or [Apex](#).
- When you delete a custom field, all of the field history data is deleted and changes are no longer tracked.
- A background process periodically runs that cleans up metadata associated with deleted custom fields. This process will affect the `Last Modified Date` and `Last Modified By` fields on custom objects.

Managing Deleted Custom Fields

User Permissions Needed	
To restore deleted custom fields and relationships:	“Customize Application”
To permanently delete custom fields or relationships:	“Customize Application”

Deleted custom fields and their data are stored until your organization permanently deletes them or 45 days has elapsed, whichever happens first. Until that time, you can restore the field and its data. However, the field still counts against the maximum number of custom fields allowed in your organization.

To view a list of your deleted custom fields and relationships:

1. Click **Create > Objects**, and click the name of the custom object in the list.
2. Click **Deleted Fields** at the bottom of the list of custom fields and relationships. The number in parentheses indicates the total number of deleted custom fields for this object. This link only displays when you have a deleted custom field.

3. Use the list of deleted fields to perform the following actions:

- To view details about a field, click the field label.
- To permanently remove the custom field and its data, click **Erase**.
- To restore the field and its data, click **Undelete**. Some attributes of deleted fields are not restored automatically. To restore these attributes manually:
 - a. [Make the field required](#) if necessary. Database.com automatically removes the required attribute for any deleted custom field.
 - b. Convert any [lookup](#) relationships to [master-detail](#) relationships if necessary. Database.com converts all relationships to lookup relationships when they are deleted. To convert a lookup relationship to a master-detail relationship, populate all the applicable records with the appropriate data.
 - c. [Redefine any field dependencies](#) that Database.com removed when the field was deleted.
 - d. [Edit and save any formula fields](#), prompting a syntax check that you may have to resolve.
 - e. [Set up field history tracking](#) if necessary. If the list of fields enabled for history tracking has been modified during the time the custom field was deleted, the restored field is no longer set up to track field history.

Notes on Restored Custom Fields

- The following characters are appended to the end of a custom field's developer name when it is deleted unless a deleted field already has that developer name: “_del.” These characters remain when you restore the custom field.
- Formula fields are restored in a disabled state, which means they do not contain updated data until you edit and save them. While a formula field is disabled, “#Error!” displays in place of the formula value.
- Restored fields do not display in search results immediately after you restore them. It may take a short time before the restored custom field and its data are available in search results.
- Auto number fields continue to increment after they are deleted and contain the correct values when restored.
- Field history data for the deleted custom field is restored.

Additional Custom Field Options

Tracking Custom Field History

You can select which custom fields to track on the History-related list of custom objects. All entries include the date, time, nature of the change, and who made the change. History data does not count against your organization’s storage limit. See [Tracking Field History](#) on page 242.

Tracking Field History

User Permissions Needed	
To set up which fields are tracked:	“Customize Application”

You can select certain custom fields to track the history of related list of custom objects. Modifying any of these standard or custom fields adds a new entry to the History related list. All entries include the date, time, nature of the change, and who made the change. History data does not count against your organization’s storage limit. Note that not all fields types are available for history tracking. .

For more information on tracking field history, see the following:

- [Tracking Field History for Custom Objects](#)
- [History Tracking Implementation Tips](#)

To set up field history tracking:

1. Click **Customize** and select the object you want to configure.
2. Click **Fields**.
3. Click **Set History Tracking**.
 - When you choose the fields you want to track, Database.com begins tracking history from that date and time forward. Changes made before that date and time are not included.
4. Choose the fields you want tracked.
5. Click **Save**.

Tracking Field History for Custom Objects

To track field history for custom objects:

1. Click **Create > Objects** and click **Edit** next to the name of the custom object.
2. Select the **Track Field History** checkbox. Deselect the checkbox if you do not want to track any changes. If you deselect the checkbox, the History related list is automatically removed from the custom object's page layouts.
3. Click **Save**.
4. Select the name of the custom object.
5. Click **Set History Tracking** in the Custom Fields & Relationships section. This section allows you to set a custom object's history tracking for both standard and custom fields.

When you choose the fields you want to track, Database.com begins tracking history from that date and time forward. Changes made before that date and time are not included.

If you deselected the **Track Field History** checkbox, the **Set History Tracking** button does not display.

6. Choose the fields you want tracked.
7. Click **Save**.

History Tracking Implementation Tips

Administration

- You can select a combination of up to 20 custom fields per object.
- You cannot track the following fields:
 - ◊ History of formula, roll-up summary, or auto-number
 - ◊ Created By and Last Modified By
- Field history is stored for 18 months.
- To archive field history, you can:
 - ◊ Schedule a regular export of **FieldHistory** data
 - ◊ Run a query using the [Web services API](#) and save your results
- If you disable field history tracking on an object, you can still report on its history data up to the date and time you disabled tracking.
- You cannot disable field history tracking for an object if a field on the object is referenced in an Apex script. For more information, see [Apex Code Overview](#) on page 379.
- If the parent record in a lookup relationship is deleted, the field history tracking for the child record does not record the deletion.

Customization

- You cannot customize the History related list because it does not store data. The History related list links to data stored elsewhere.
- When you delete a custom field, all of the field history data is deleted and changes are no longer tracked.
- If you disable field history tracking on a custom object, then you cannot report on its field history.

Management

- Changes to fields with more than 255 characters are tracked as edited, and their old and new values are not recorded. For example, changes to long text area fields are tracked as edited.
- Changes to date fields, number fields, and standard field labels are shown in the locale of the user viewing the History related list. For example, a date change to August 8, 2005 shows as 8/5/2005 for a user with the English (United States) locale and as 5/8/2005 for a user with the English (United Kingdom) locale.
- [Field updates](#) are tracked in the History related list if you have set history tracking on those fields.

Viewing Fields

To view the details for a custom field:

1. Click **Create > Objects**, and click the name of the custom object in the list.
2. Click the name of the field.
3. From the field detail page, you can:
 - Click **Edit** to modify a custom field or change the custom field's data type. See [Changing Custom Field Type](#) on page 113.
 - Click **Set Field-Level Security** to [set users' access to the field](#).
 - Click **View Field Accessibility** to [view who can access the field based on permissions](#).
 - If the field is a dependent picklist, view a list of the controlling fields on which it depends. Click **New** to define a dependent picklist. Click **Edit** to change the picklist dependency rules or **Del** to remove the picklist dependency. For information on dependent picklists, see [Defining Dependent Picklists](#) on page 188.
 - If the custom field is a dependent picklist, click **[Change]** next to the controlling field to edit the picklist dependency rules. For information on dependent picklists, see [Defining Dependent Picklists](#) on page 188.
 - Click **New** in the Validation Rules related list to create a new validation rule for the field. Click **Edit** to change a validation rule or **Del** to remove it. See [Defining Validation Rules](#) on page 175.
 - For picklist fields, you can add new values and edit, replace, or delete existing values. See “Additional Options for Updating Picklists” in the online help.
 - To change External ID, Required, or other attributes under the General Options section, see [Custom Field Attributes](#) on page 101.
 - Click **Undelete** to restore the field and its data. This is only available if the field has been deleted but not permanently erased. The field's behavior may be different after restoring it. To restore the field completely, see [Managing Deleted Custom Fields](#) on page 106.

Merge Fields Overview

You can use merge fields within custom formula fields. Merge fields serve as placeholders for data that will be replaced with information from your records, user information, or company information.

Guidelines for Using Merge Fields

- [General merge field syntax guidelines](#):

The syntax for a merge field may differ, depending on where you are using it in Database.com. Here are a few examples of syntax differences:

- ◊ For formula fields, the merge field is not enclosed in curly braces or preceded by an exclamation point, nor is it preceded by the type of record, for example: AccountNumber.
- ◊ For custom objects, the field label is preceded with the type of record and all spaces are converted to underscores. For example: {!Stockforce_CreatedDate} references the standard field called Created Date for the Stockforce custom object.
- ◊ For formulas that allow you to reference fields on related objects across multiple relationships, the field name is prefixed by the name of the relationship. For custom relationships, the name of the relationship is the Field Name given when creating the relationship with “_r” appended to it.
- ◊ Database.com rounds numbers referenced in merge fields according to the user's locale, not the number of decimal of spaces specified in the number field configuration.

Managing Custom Field Types

Custom Field Types

The first step in creating a custom field is choosing the type of the field. Below is a description of each custom field type.

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.
Checkbox	Allows users to check a box, indicating a true or false attribute of a record. When using a checkbox field for list view filter, use “True” for checked values and “False” for unchecked values.
Currency	Allows users to enter a currency amount. The system automatically formats the field as a currency amount. This can be useful if you export data to Excel or another spreadsheet.  Note: Database.com uses the round half even tie-breaking rule for currency fields. For example, 23.5 becomes 24, 22.5 becomes 22, -22.5 becomes -22, and -23.5 becomes -24. Values lose precision after 15 decimal places.
Date	Allows users to enter a date or pick a date from a popup calendar.
Date/Time	Allows users to enter a date or pick a date from a popup calendar and enter a time of day. They can also add the current date and time by clicking the date and time link next to the field. The time of day includes AM or PM notation. In reports, you can limit the data by specific dates and times using any custom date field.
Email	Allows users to enter an email address, which is validated to ensure proper format.

Type	Description
Formula	<p>Allows users to automatically calculate values based on other values or fields such as merge fields. See Building Formulas on page 117 and Operators and Functions on page 132.</p> <p> 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.</p>
Hierarchical Relationship	<p>Creates a hierarchical lookup relationship between users. Allows users to use a lookup field to associate one user with another that does not directly or indirectly refer to itself. For example, you can create a custom hierarchical relationship field to store each user's direct manager.</p>
Lookup Relationship	<p>Creates a relationship between two records so you can associate them with each other. A lookup relationship creates a field that allows users to click a lookup icon and select another record from a popup window. On the associated record, you can then display a related list to show all of the records that are linked to it. You can create lookup relationship fields that link to users or custom objects. A lookup relationship has no effect on record deletion or security, and the lookup field is not required. If a lookup field references a record that is deleted, Database.com sets the lookup field to <code>null</code>, and does not run any Apex triggers, validation rules, workflow rules, or roll-up summary fields. For more information on relationships, see Overview of Relationships on page 91</p>
Master-Detail Relationship	<p>Creates a relationship between records where the master record controls certain behaviors of the detail record such as record deletion and security. For more information on relationships, see Overview of Relationships on page 91.</p>
Number	<p>Allows users to enter any number. This is treated as a real number and any leading zeros are removed.</p> <p> Note: Database.com uses the round half up tie-breaking rule for number fields. For example, 12.345 becomes 12.35 and -12.345 becomes -12.34. Database.com rounds numbers referenced in merge fields according to the user's locale, not the number of decimal or spaces specified in the number field configuration.</p>

Type	Description
Percent	<p>Allows users to enter a percentage number, for example, '10'. The system automatically adds the percent sign to the number.</p> <p> Note: If the decimal value is greater than 15, and you add a percent sign to the number, a runtime error occurs.</p> <p>Values lose precision after 15 decimal places.</p>
Phone	Allows users to enter any phone number.
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	Allows users to enter any combination of letters, numbers, or symbols. You can set a maximum length, up to 255 characters.
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 salesforce.com. See also Managing Database.com Master Encryption Keys on page 298.
Text Area	Allows users to enter up to 255 characters that display on separate lines similar to a Description field.
Text Area (Long)	Allows users to enter up to 32,768 characters that display on separate lines similar to a Description field. You can set the length of this field type to a lower limit, if desired. Any length from 256 to 32,768 characters is allowed. Note that every time you press Enter within a long text area field, a linebreak and a return character are added to the text. These two characters count toward the 32,768 character limit.

Type	Description
Text Area (Rich)	With the use of a toolbar, users can format the field content and add images and hyperlinks. The toolbar allows the user to undo, redo, bold, italicize, underline, strike-out, add a hyperlink, upload or link to an image, modify alignment, add a numbered or non-numbered list, indent, and outdent. The maximum field size is 32,768 characters, inclusive of all the formatting and HTML tags. The maximum size for uploaded images is 1MB. Only gif, jpeg and png file types are supported.
URL	Allows users to enter up to 255 characters of any valid website address. When users click on the field, the URL will open in a separate browser window. Note that only the first 50 characters are displayed on the record detail pages.

Changing Custom Field Type

User Permissions Needed	
To change custom fields:	“Customize Application”

To change the data type of an existing custom field:

1. Click **Create > Objects**, and click the name of the custom object in the list.
2. Click **Edit** next to the custom field you want to change.
3. Click **Change Field Type**.
4. Select a new data type and click **Next**.
5. Enter a field label, name, any other attributes, and click **Save**.

For more information, see [Notes on Changing Custom Field Types](#) on page 113.

Notes on Changing Custom Field Types

Consider the following before converting fields:

- Only convert custom fields for which no data exists or you risk losing your data. Changing the data type of an existing custom field can cause data loss in the following situations:
 - ◊ Changing to or from type Date or Date/Time
 - ◊ Changing to Number from any other type
 - ◊ Changing to Percent from any other type
 - ◊ Changing to Currency from any other type
 - ◊ Changing from Checkbox to any other type
 - ◊ Changing from Picklist (Multi-Select) to any other type
 - ◊ Changing to Picklist (Multi-Select) from any other type

[Currently defined picklist values](#) are retained when you change a picklist to a multi-select picklist. If records contain values that are not in the picklist definition, those values will be deleted from those records when the data type changes.

- ◊ Changing from Auto Number to any other type
 - ◊ Changing to Auto Number from any type except Text
 - ◊ Changing from Text Area (Long) to any type except Email, Phone, Text, Text Area, or URL
- If data is lost, any list view based on the custom field will be deleted, and assignment and escalation rules may be affected.
 - If you change the data type of a custom field that is set as an external ID, choosing a data type other than text, number, or email will cause the field to no longer act as an external ID.
 - The option to change the data type of a custom field is not available for all data types. For example, existing custom fields cannot be converted into [encrypted fields](#) nor can encrypted fields be converted into another data type.
 - For descriptions of other attributes you can set, see [Custom Field Attributes](#) on page 101.
 - Changing a custom field type may require changing a large number of records at once.

The following data types have additional restrictions when you convert them:

Data Type	Description
Auto Number	The data in any auto-number field remains unchanged if you convert it into a text field. Also, you can safely convert a text custom field into an auto-number field without losing your data. Converting an auto-number field into any other data type results in data loss. Auto-number fields can contain a maximum of 30 characters. Before converting a text custom field into an auto-number field, change any records that contain more than 30 characters in that field.
Formula	Formula fields are special read-only fields that cannot be converted to any other data type. Likewise, you cannot convert any other field type into a formula field.
Picklist	Changing your custom picklists into custom checkboxes is simple. If you select Checkbox as the new data type, you can choose which picklist values to map to checked boxes and unchecked boxes. You can change custom picklists into multi-select picklists without losing any data. Since your records only contain a single value of that picklist, that value will still be selected but users can select additional values. You can also change a picklist custom field into a text custom field or a text custom field into a picklist custom field without any data loss.
Relationships	<ul style="list-style-type: none"> • If your organization has a large number of records, Database.com displays a waiting page after you have requested to change a master-detail into a lookup relationship or a lookup into a master-detail relationship. • After you have created a roll-up summary field on an object, you cannot convert the object's master-detail relationship into a lookup relationship. • A lookup cannot be converted to a master detail relationship if there are any existing records on the object that have a null value set for the lookup relationship.

Data Type	Description
Text Area (Long)	When you convert a long text area field to an Email, Phone, Text, Text Area, or URL type field, the data in your records is truncated to the first 255 characters of the field.
Text Area (Rich)	You can only convert rich text area fields into long text area fields. Any images are deleted the next time the long text area field is saved.



Note: You cannot change the data type of a custom field if it is referenced in an Apex script. For more information, see [Apex Code Overview](#) on page 379.

Managing Default Field Values

About Default Field Values

Use default field values to make your users more productive by reducing the number of fields they need to fill in manually. Default field values automatically insert the value of a custom field when a new record is created. A default value can be based on a formula for some types of fields or exact values such as Checked or Unchecked for checkbox fields.

The user can change the field's value but the initial default field value is only executed once, during record creation. You can change this value later, but you cannot automatically restore the value that was seven days after the creation date.

Set up default field values for the following types of custom fields:

- Checkbox
- Currency
- Date
- Date/Time
- Email
- Number
- Percent
- Phone
- Picklist (use the default option when setting up the picklist)
- Text
- Text Area
- URL

For a description of these types, see [Custom Field Types](#) on page 110.

Default Field Value Considerations

Default field values automatically insert the value of a custom field when a new record is created. A default value can be based on a formula for some types of fields or exact values such as Checked or Unchecked for checkbox fields. Review the following considerations before incorporating default field values in your organization.

- If a default value is based on the value of a [merge field](#), Database.com uses the value of the merge field at the time the default value is executed. If the value of the merge field changes later, the default value is not updated.
- Users can change or remove the default field value on a record.

- **Default values** should not be assigned to fields that are both **required** and **unique**, as uniqueness errors may result.
- Default field values are different from formula fields in the following ways: they are only executed once, at record creation; they are not read only; and the user can change the value but cannot restore the default field value.
- Since the default value is inserted before users enter any values in the new record, you cannot use the fields on the current record to create a default field value.
- Fields that are not visible to the user due to field-level security are still available in the formula for a default field value.



Note: You can define a formula for default values only where appropriate. For example, the default value options for picklist and checkbox fields are limited to the options available for those types of fields, such as **Checked**, **Unchecked**, or **Use first value as default value**.

Managing Formulas

About Formulas

A formula is an algorithm that derives its value from other fields, expressions, or values. Formulas can help you automatically calculate the value of a field based on other fields. Use formulas for:

- **Custom Fields**

Create custom formula fields that automatically calculate a value based on other values, merge fields, or expressions. Users can view formula fields on record detail pages but cannot see the underlying algorithm nor can they edit the value of a formula field. To create a custom formula field, see [Building Formulas](#) on page 117.

- **Default Field Values**

Apply a value to a custom field when a user creates a record. Use formulas to define a default value such as `TODAY() + 7`. See [About Default Field Values](#) on page 115.

- **Data Validations**

Verify that the data a user enters in a record meets the standards you specify before the user can save the record. A validation rule can include a formula such as `CloseDate >= TODAY()`. See [About Validation Rules](#) on page 172.

- **Workflow Field Updates**

Define field updates to automatically calculate the new value of a field based on a formula. The formula can include other values, merge fields, or expressions. Then set your workflow rules or approval processes to use these field updates. To define field updates, see [Defining Field Updates](#) on page 416.

- **Workflow Rules**

Define the criteria a record must meet to trigger a workflow rule. See [Creating Workflow Rules](#) on page 413.

Before building a formula, review the following attributes of a formula:

- [Formula Data Types](#)
- [Elements of a Formula](#)

For common uses of formulas, see “Formulas: How Do I...” in the online help.

What are Cross-Object Formulas?

User Permissions Needed	
To create or change cross-object formulas:	“Customize Application”

Cross-object formulas are formulas that span two related objects and reference merge fields on those objects. Cross-object formulas can reference merge fields from a master (“parent”) object if an object is on the detail side of a master-detail relationship. You can reference fields from objects that are up to ten relationships away. Cross-object formulas are available anywhere formulas are used except when creating default values.

To create a cross-object formula when building a formula in the **Simple Formula** tab, enter the relationship names of the objects to which you are spanning followed by the field you want to reference. Separate the relationship names of each object and the field with periods.

To create a cross-object formula when building a formula in the **Advanced Formula** tab or for rules, such as workflow, validation, assignment, auto-response, or escalation rules, click **Insert**, then click the related object to list its fields. Related objects are denoted by a “>” sign.



Note: The value of the `Profile.Name` merge field differs depending on the context of the cross-object formula field that references it. On detail pages, the value is the profile name, as expected; however, in list views, the value is the internal value of the associated profile instead. If you use `Profile.Name` in a formula, use it within an `OR` function to ensure the formula always returns the intended result. For example:

```
IF
  (OR
    (LastModifiedBy.Profile.Name = "Standard User", LastModifiedBy.Profile.Name
     = "PT2"),
    "Standard", "Not Standard")
```

None of the above applies to profile names referenced by the `$Profile` global variable.

Building Formulas

User Permissions Needed	
To view formula field details:	“View Setup and Configuration”
To create, change, or delete formula fields:	“Customize Application”

Your custom formula fields require special attributes. To build your formula:

1. Begin building a formula field the same way you create a custom field. See [Creating Custom Fields](#) on page 98.
2. Select the data type for the formula. Choose the appropriate data type for your formula based on the output of your calculation. See [Formula Data Types](#) on page 119.
3. Choose the number of decimal places for currency, number, or percent data types. This setting is ignored for currency fields in multicurrency organizations. Instead, the `Decimal Places` for your currency setting apply.



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.

4. Click **Next**.
5. Build your formula:

- a. If you are building a formula in the **Advanced Formula** tab or for workflow or validation rules, click **Insert Field**, choose a field, and click **Insert**.

To create a basic formula that passes specific Database.com data, select the **Simple Formula** tab, choose the field type in the **Select Field Type** drop-down list, and choose one of the fields listed in the **Insert Field** drop-down list.



Tip: Build *cross-object formulas* to span to related objects and reference merge fields on those objects.

- b. To insert an operator, choose the appropriate operator icon from the **Insert Operator** drop-down list. Use the examples in [Operators and Functions](#) on page 132.
- c. Optionally, click the **Advanced Formula** tab to use functions and view additional operators and merge fields. Functions are prebuilt formulas that you can customize with your input parameters. For a description of each operator and function, see [Operators and Functions](#) on page 132.
- d. To insert a function, double-click its name in the list, or select it and click **Insert Selected Function**. To filter the list of functions, choose a category from the **Functions** drop-down list. Select a function and click **Help on this function** to view a description and examples of formulas using that function.
- e. Consider adding comments to your formula, especially if it is complicated. Comments must begin with a forward slash followed by an asterisk (/ *), and conclude with an asterisk followed by a forward slash (* /).

Comments are useful for explaining specific parts of a formula to anyone viewing the formula definition.

You can also use comments to *comment out* sections of your formula when debugging and checking the syntax to locate errors in the formula.



Note:

- Nesting comments causes a syntax error. For example, you cannot save a formula that has the following:

```
/* /* comment */ */
```

- Commenting out a whole formula causes a syntax error.
- Comments count against the character and byte size limits in formulas.



Note: Formula fields can contain up to 3,900 characters, including spaces, return characters, and comments. If your formula requires more characters, create separate formula fields and reference them in another formula field. The maximum number of displayed characters after an evaluation of a formula expression is 1,300 characters.

6. Click **Check Syntax** to check your formula for errors.
7. Optionally, enter a description of the formula in the **Description** box.
8. If your formula references any number, currency, or percent fields, choose an option for handling blank fields. To give any blank fields a zero value, choose **Treat blank fields as zeros**. To leave these fields blank, choose **Treat blank fields as blanks**.
9. Click **Next**.
10. Set the field-level security to determine whether the field should be visible for specific profiles, and click **Next**.
11. Click **Save** to finish or **Save & New** to create more custom fields.



Note: Because formula fields are automatically calculated, they are read-only on record detail pages and do not update last modified date fields. Formula fields are not visible on edit pages.

Formula Data Types

The data type of a formula determines the type of data you expect returned from your formula. Review the following data types.

Data Type	Description
Currency	Returns a number in currency format of up to 18 digits with a currency sign.  Note: Database.com uses the round half even tie-breaking rule for currency fields. For example, 23.5 becomes 24, 22.5 becomes 22, -22.5 becomes -22, and -23.5 becomes -24.
Date	Returns data that represents a day on the calendar. The current date can be acquired by calling the built-in function TODAY() in a formula.
Date/time	Returns data that represents a moment in time. A date/time field includes the date and also the time of day including hour, minutes, and seconds. You can insert the current date and time in a formula using the NOW() function.
Number	Returns a positive or negative integer or decimal of up to 18 digits. 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.
Percent	Returns a number in percent format of up to 18 digits followed by a percent sign. Percent data is stored as a decimal divided by 100, which means that 90% is equal to 0.90.
Text	Returns a string of up to 3900 characters. To display text in addition to the formula output, insert that text in quotes. Use the text data type for text, text area, URL, phone, email, address, and auto-number fields.

Elements of a Formula

A formula can contain references to the values of fields, operators, functions, literal values, or other formulas. Use any or all of the elements in the following table to build a formula. For more details about building formulas, see [Building Formulas](#) on page 117.

Element Name	Description
Literal Value	A text string or number you enter that is not calculated or changed. For example, if you have a value that's always

Element Name	Description
	<p>multiplied by 2% of an amount, your formula would contain the literal value of 2% of that amount:</p> <pre data-bbox="840 361 1171 388">ROUND((Amount*0.02) , 2)</pre> <p>This example contains every possible part of a formula:</p> <ul style="list-style-type: none"> • A function called ROUND used to return a number rounded to a specified number of decimal places. • A field reference called Amount. • An operator, *, that tells the formula builder to multiply the contents of the Amount field by the literal value, 0.02. • A literal number, 0.02. Use the decimal value for all percents. To include actual text in your formula, enclose it in quotes. • The last number 2 in this formula is the input required for the ROUND function that determines the number of decimal places to return.
Field Reference	<p>The syntax for a merge field on a related object is <code>object_name__r.field_name</code>. Use the Insert Field button or the drop-down list to insert a merge field in your formula where necessary.</p>
Function	<p>A system-defined formula that can require input from you and returns a value or values. For example, TODAY() does not require input but returns the current date. The TEXT(value) function requires your percent, number, or currency input and returns text.</p>
Operator	<p>A symbol that specifies the type of calculation to perform or the order in which to do it. For example, the + symbol specifies two values should be added. The open and close parentheses specify which expressions you want evaluated first.</p>
Comment	<p>An annotation within a formula that begins with a forward slash followed by an asterisk /*). and concludes with an asterisk followed by a forward slash (*/). For example,</p> <pre data-bbox="840 1558 1263 1586">/*This is a formula comment*/</pre> <p>Comments are ignored when processing a formula. Comments are useful for explaining specific parts of a formula to anyone viewing the formula definition.</p> <p>You can also use comments to <i>comment out</i> sections of your formula when debugging and checking the syntax to locate errors in the formula.</p>

Element Name	Description
	<p> Note:</p> <ul style="list-style-type: none"> Nesting comments causes a syntax error. For example, you cannot save a formula that has the following: <pre>/* /* comment */ */</pre> Commenting out a whole formula causes a syntax error. Comments count against the character and byte size limits in formulas.

Tips on Building Formulas

- Formula fields that a user can see may reference fields that are hidden or read only using field-level security. If the formula field contains sensitive information, use field-level security to hide it. See [Field-Level Security Overview](#) on page 264.
- The following limits apply to formula fields:
 - ◊ Character limit—Formula fields can contain up to 3,900 characters, including spaces, return characters, and comments. If your formula requires more characters, create separate formula fields and reference them in another formula field.
 **Note:** The maximum number of displayed characters after an evaluation of a formula expression is 1,300 characters.
 - ◊ Save size limit—Formula fields cannot exceed 4,000 bytes when saved. The save size differs from the number of characters if you use multi-byte characters in your formula.
 - ◊ Compile size limit—Formula fields cannot exceed 5,000 bytes when compiled. The compile size is the size of the formula (in bytes) including all of the fields, values, and formulas it references. There is no direct correlation between the compile size and the character limit. Some functions, such as TEXT, DATEVALUE, DATETIMEVALUE, and DATE significantly increase the compile size.
- Long text area, encrypted, and Description fields are not available for use in formulas.
- The value of a field cannot depend on another formula that references it.
- Fields referenced in formulas cannot be deleted. Remove the field from the formula before deleting it.
- Use the `Istask` merge field to determine if a record is a task or event. For example, the following formula displays text indicating if the record is a task or event:

```
IF(Istask, "This is a task", "This is an event")
```

- To reference the unique identifier for your Database.com organization in a formula, insert the `$Organization.Id` merge field. This merge field can display anywhere formula fields can.
- Some merge fields display as radio buttons but function like picklist fields when referenced in a formula.

Working with Date and Date/Time Fields

- Dates and times are always calculated using the user's time zone.
- Date and date/time fields cannot be used interchangeably. The name alone may not indicate if a field is a date or date/time. For example, `Created Date` and `Last Modified Date` are date/time fields whereas `Last Activity Date` is a date field. Use the [DATEVALUE](#) function to convert a date/time field into a date field.
- Use addition and subtraction operators with date or date/time fields to calculate duration. For example, subtract a date from another date to calculate the number of days between the two. Likewise, you can subtract the date/time from another date/time to get the number of days between the two in the form of a number. See [NOW](#) on page 162 or [TODAY](#) on page 169 for suggested use.
- Use addition and subtraction operators with numbers to return another date or date/time. For example, `{ !CreatedDate } + 5` calculates the date and time five days after a record's created date. Note that the expression returns the same data type as the one given; a date field plus or minus a number returns a date, and a date/time field plus or minus a number returns a date/time.
- When calculating dates using fractions, Database.com ignores any numbers beyond the decimal. For example:

`TODAY() + 0.7` is the same as `TODAY() + 0`, which is today's date

`TODAY() + 1.7` is the same as `TODAY() + 1`, which is tomorrow's date

`TODAY() + (-1.8)` is the same as `TODAY() + (-1)`, which is yesterday's date

- To calculate the value of two fractions first, group them within parentheses. For example:
- `TODAY() + 0.5 + 0.5` is the same as `TODAY() + 0 + 0`, which is today's date
- `TODAY() + (0.5+0.5)` is the same as `TODAY() + 1`, which is tomorrow's date

- Years cannot be zero and must be between -4713 and 9999.

Working with Text Fields

- Before using the [HYPERLINK](#) function, consider these:
 - ◊ Hyperlink formula fields open in a new browser window by default or you can specify a different target window or frame.
 - ◊ Hyperlink formula fields that contain relative URLs to Database.com pages can be added to list views and related lists. However, use a complete URL, including the server name and `https://`, in your hyperlink formula before adding it to a search layout. Note that formula fields are not available in search result layouts.
- To insert text in your formula field, surround the text with quotation marks.
- Use the backslash (\) character before a quote or backslash to insert it as a literal value in your output.

Working with Number Fields

- Use the decimal version of a percent when working with percent fields in formulas. For example, `IF(Probability =1...)` for 100% probability or `IF(Probability =0.9...)` for 90% probability.
- Reference auto-number fields as text fields in formulas.
- The output of your formula must be less than 19 digits. See [Formula Errors](#) on page 123.
- Formulas can contain a mix of numbers, percents, and currencies as in this example: `AnnualRevenue / NumberOfEmployees`.
- 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.

Working with Cross-Object Formulas

- Cross-object formulas that reference currency fields convert the value to the currency of the record that contains the formula.

- Database.com allows a maximum of ten unique relationships per object in cross-object formulas. The limit is cumulative across all formula fields, rules, and lookup filters.
- You cannot reference cross-object formulas in roll-up summary fields.
- Do not use the \$RecordType global variable in cross-object formulas. The \$RecordType variable only resolves to the record containing the formula, not the record to which the formula spans.

Working with Picklists and Multi-Select Picklists

- Picklist fields can only be used in the following functions:
 - ◊ **ISPICKVAL**—Compares the value of a picklist to a single value.
 - ◊ **CASE**—Compares the value of a picklist to multiple values.
 - ◊ **TEXT**—Converts a picklist value into a text value so that you can work with the value in functions that support text value, such as CONTAINS. (Only available in formula fields, validation rules, and workflow field updates.)
- The TEXT function always returns picklist values in your organization's master language, not the language of the current user.
- Multi-select picklist fields can only be used in the following functions:
 - ◊ **INCLUDES**
 - ◊ **ISBLANK**
 - ◊ **ISNULL**
 - ◊ **ISCHANGED** (Only in assignment rules, validation rules, workflow field updates, and workflow rules in which the trigger type is set to **Every time a record is created or edited**)
 - ◊ **PRIORVALUE** (Only in assignment rules, validation rules, workflow field updates, and workflow rules in which the trigger type is set to **Every time a record is created or edited**)

Formula Errors

User Permissions Needed	
To view formula field details:	"View Setup and Configuration"
To create, change, or delete formula fields:	"Customize Application"

- “#Error!” displays for a formula field whenever an error occurs while calculating the value of a formula. To resolve the error, check your formula.
 - ◊ Is the formula dividing by zero? If so, check if the denominator of your expression is zero and provide an alternative value.
 - ◊ Is the formula calculating a value larger than the maximum value of the current type? If so, you can append L to numeric values to make them Long so the intermediate products will be Long and no overflow occurs. For example, the following example shows how to correctly compute the amount of milliseconds in a year by multiplying Long numeric values.

```
Long MillsPerYear = 365L * 24L * 60L * 60L * 1000L;
Long ExpectedValue = 31536000000L;
System.assertEquals(MillsPerYear, ExpectedValue);
```

- ◊ Is the formula calculating the square root of a negative number? If so, use an IF function similar to the one above to check if the value is a positive number.

- ◊ Is the formula calculating the LOG of a negative number? If so, use an IF function similar to the one above to make sure that the number is positive.
- ◊ Is the formula using the VALUE function with text that contains special characters? For examples of special characters, see [Operators and Functions](#) on page 132.
- ◊ Make sure the formula does not contain a HYPERLINK function within a text function, such as LEFT(HYPERLINK("http://MYCOMPANY.ORG ", "MYCOMPANY ") , 5).
- ◊ Is the formula disabled or referencing a disabled formula field? Database.com disables formula fields when they are deleted and they remain disabled after they are restored. To enable disabled formula fields, edit and save the field. For more information on deleted custom fields and restoring them, see [Managing Deleted Custom Fields](#) on page 106.
- “#Too Big!” displays if your formula output is over 18 digits. When this happens, check your formula for calculations that could result in more than 18 digits. Avoid multiplying large numbers, raising a large number to a power, or dividing by a very small number.
- Prevent division by zero errors by including an IF function that determines if the value of a field is zero. For example, IF(Field_c =0, 0, 25/Field_c).

Examples of Advanced Formula Fields

User Permissions Needed	
To view formula field details:	“View Setup and Configuration”
To create, change, or delete formula fields:	“Customize Application”

Use the following formula samples when creating custom formula fields.

This document contains the following categories of custom formula samples:

- [Data Categorization](#)
- [Date Calculations](#)
- [Expense Tracking](#)
- [Financial Calculations](#)
- [Image Links](#)
- [Integration Links](#)
- [Metrics](#)
- [Project Management](#)

Account Media Service Links

BBC™ News Search

This formula creates a link to a BBC news search site based on the Account Name.

```
HYPERLINK(
"http://newssearch.bbc.co.uk/cgi-bin/search/results.pl?scope=newsifs;tab=news;q=&Name,
"BBC News")
```

Bloomberg™ News Search

This formula creates a link to an account's ticker symbol on the Bloomberg website.

```
HYPERNLINK(
"http://www.bloomberg.com/apps/quote?ticker=&TickerSymbol,
"Bloomberg News")
```

CNN™ News Search

This formula creates a link to a CNN news search site using the Account Name.

```
HYPERNLINK(
"http://websearch.cnn.com/search/search?source=cnn&
invocationType=search%2Ftop&sites=web&query=&Name,
"CNN News")
```

MarketWatch™ Search

This formula creates a link to an account's ticker symbol on the Marketwatch.com website.

```
HYPERNLINK(
"http://www.marketwatch.com/tools/quotes/quotes.asp?symb=&TickerSymbol,
"Marketwatch")
```

Google™ Search

This formula creates a link to a Google search site using the Account Name.

```
HYPERNLINK(
"http://www.google.com/search?en&q=&Name,
"Google")
```

Google News Search

This formula creates a link to a Google news search site using the Account Name.

```
HYPERNLINK(
"http://www.google.com/news?en&q=&Name,
"Google News")
```

Yahoo!™ Search

This formula creates a link to a Yahoo! search site using the Account Name.

```
HYPERNLINK(
"http://search.yahoo.com/search?p=&Name,
"Yahoo Search")
```

Yahoo! News Search

This formula creates a link to a Yahoo! news search site using the Account Name.

```
HYPERNLINK(
"http://news.search.yahoo.com/search/news?p=&Name,
"Yahoo News")
```

For details about using the function used in these formulas, see [HYPERLINK](#) on page 148.

Data Categorization

Deal Size Large and Small

This formula displays “Large Deal” for deals over one million dollars or “Small Deal” for deals under one million dollars.

```
IF(Sales_Price__c > 1000000,  
"Large Deal",  
"Small Deal")
```

For details about using this function, see [IF](#) on page 149.

Deal Size Small

This formula displays “Small” if the price and quantity are less than one. This field is blank if the asset has a price or quantity greater than one.

```
IF(AND(Price<1,Quantity<1),"Small",  
null)
```

For details about using these functions, see [IF](#) on page 149 and [AND](#) on page 141.

Product Categorization

This formula checks the content of a custom text field named `Product_Type` and returns “Parts” for any product with the word “part” in it. Otherwise, it returns “Service.” Note that the values are case sensitive, so if a `Product_Type` field contains the text “Part” or “PART,” this formula returns “Services.”

```
IF(CONTAINS(Product_Type__c, "part"), "Parts", "Service")
```

For details about using these functions, see [IF](#) on page 149 and [CONTAINS](#) on page 144.

Date Calculations

Birthday in Current Year Accounting for Leap Years

This formula returns the date of a person's birthday in the current year, even if the person's birthday is on February 29th in a leap year.

```
IF(AND(MONTH(Birthdate) = 2, DAY(Birthdate) = 29),  
(IF(OR(MOD(YEAR(DATEVALUE(NOW())), 400) = 0, AND(MOD(YEAR(DATEVALUE(NOW())), 4) = 0,  
MOD(YEAR(DATEVALUE(NOW())), 100) <> 0)),  
DATE(YEAR(DATEVALUE(NOW())), MONTH(Birthdate), DAY(Birthdate)),  
DATE(YEAR(DATEVALUE(NOW())), MONTH(Birthdate + 1), 28)),  
(DATE(YEAR(DATEVALUE(NOW())), MONTH(Birthdate), DAY(Birthdate))))
```

Day of Week (number)

This formula calculates today's day of the week as a number (0 = Sunday, 1 = Monday, 2 = Tuesday, and so on).

```
MOD(TODAY() - DATE(1900, 1, 7), 7)
```

Similarly, this formula substitutes the `TODAY()` function shown in the previous example with a custom date field called `Sign Up Date`. It returns the day of the week as a number for that field.

```
MOD(Sign_Up_Date__c - DATE(1900, 1, 7), 7)
```

For details about using these functions, see [MOD](#) on page 161, [TODAY](#) on page 169, and [DATE](#) on page 145.

Day of Week (text)

This formula calculates today's day of the week and displays it as text. To determine the day of the week for a date field, use the formula below and replace "TODAY()" with that date field.

```
CASE (
  MOD(TODAY() - DATE(1900, 1, 7), 7),
  0, "Sunday",
  1, "Monday",
  2, "Tuesday",
  3, "Wednesday",
  4, "Thursday",
  5, "Friday",
  6, "Saturday", "Error")
```

For details about using these functions, see [CASE](#) on page 142, [MOD](#) on page 161, [TODAY](#) on page 169, and [DATE](#) on page 145.

Day of Year

This formula calculates today's numeric day of the year (a number between 1 and 365).

```
TODAY() - DATE(YEAR(TODAY()), 1, 1) + 1
```

For details about using these functions, see [TODAY](#) on page 169, [DATE](#) on page 145, and [YEAR](#) on page 172.

Days Until End of Month

This formula displays the number of days between a specific date and the end of the month in which the date occurs.

```
IF (
MONTH(CloseDate)=12,
DATE(YEAR(CloseDate),12,31) - CloseDate,
DATE(YEAR(CloseDate),
MONTH(CloseDate)+1,1) - CloseDate-1)
```

For details about using these functions, see [IF](#) on page 149, [MONTH](#) on page 162, [DATE](#) on page 145, and [YEAR](#) on page 172.

Time of Day

This formula returns the time of day in Greenwich Mean Time (GMT), for example: "08:04 PM".

```
MID(TEXT(Due_Date_Time_c), 12, 5)
```

For details about using these functions, see [MID](#) on page 160 and [TEXT](#) on page 168.

Expense Tracking

Expense Identifier

This formula displays the text “Expense-” followed by trip name and the expense number. This is a text formula field that uses an expense number custom field.

```
"Expense-"
& Trip_Name__c & "-" & ExpenseNum__c
```

For details about using this operator, see [- \(Subtract\)](#) on page 137.

Mileage Calculation

This formula calculates mileage expenses for visiting a customer site at 35 cents a mile.

```
Miles_Driven__c * 0.35
```

For details about using this operator, see [* \(Multiply\)](#) on page 137.

Financial Calculations

Compound Interest

This formula calculates the interest, you will have after T years, compounded M times per year.

```
Principal__c * ( 1 + Rate__c / M ) ^ ( T * M )
```

For details about using these operators, see [* \(Multiply\)](#) on page 137, [/ \(Divide\)](#) on page 137, and [^ \(Exponentiation\)](#) on page 137.

Compound Interest Continuous

This formula calculates the interest that will have accumulated after T years, if continuously compounded.

```
Principal__c * EXP(Rate__c * T)
```

For details about using this function, see [EXP](#) on page 146.

Consultant Cost

This formula calculates the number of consulting days times 1200 given that this formula field is a currency data type and consulting charges a rate of \$1200 per day. Note that Consulting Days is a custom field.

```
Consulting_Days__c
* 1200
```

For details about using this operator, see [* \(Multiply\)](#) on page 137.

Gross Margin

This formula provides a simple calculation of gross margin. In this formula example, Total Sales and Cost of Goods Sold are custom currency fields.

```
Total_Sales__c - Cost_of_Goods_Sold__c
```

For details about using this operator, see [- \(Subtract\)](#) on page 137.

Gross Margin Percent

This formula calculates the gross margin based on a margin percent.

```
Margin_percent__c * Items_Sold__c * Price_item__c
```

For details about using this operator, see [* \(Multiply\)](#) on page 137.

Payment Due Indicator

This formula returns the date five days after the contract start date whenever Payment Due Date is blank. Payment Due Date is a custom date field.

```
(BLANKVALUE(Payment_Due_Date__c, StartDate  
+5))
```

For details about using this function, see [BLANKVALUE](#) on page 141.

Payment Status

This formula determines if the payment due date is past and the payment status is “UNPAID.” If so, it returns the text “PAYMENT OVERDUE” and if not, it leaves the field blank. This example uses a custom date field called Payment Due Date and a text custom field called Payment Status on contracts.

```
IF(  
AND(Payment_Due_Date__c < TODAY(),  
ISPICKVAL(Payment_Status__c, "UNPAID")),  
"PAYMENT OVERDUE",  
null )
```

For details about using these functions, see [IF](#) on page 149, [AND](#) on page 141, [TODAY](#) on page 169, and [ISPICKVAL](#) on page 156.

Image Links

Yahoo! Instant Messenger™ Image

This formula displays an image that indicates whether a contact or user is currently logged in to Yahoo! Instant Messenger. Clicking the image launches the Yahoo! Instant Messenger window. This formula uses a custom text field called Yahoo Name to store the contact or user’s Yahoo! ID.

```
IF(ISBLANK(Yahoo_Name__c), "", HYPERLINK("ymsgr:sendIM?" & Yahoo_Name__c,  
IMAGE("http://opi.yahoo.com/online?u=" & Yahoo_Name__c & "&m=g&t=0", " ")))
```

For details about using these functions, see [IF](#) on page 149, [LEN](#) on page 158, [HYPERLINK](#) on page 148, and [IMAGE](#) on page 150.

"Skype Me™" Auto Dialer Button

This formula displays an image that looks like a push button. Clicking the button automatically dials the specified phone number.

```
HYPERLINK("callto://" & "+1" & Phone,
IMAGE("http://goodies.skype.com/graphics/skypeme_btn_small_blue.gif",
"Click to Skype"))
```

For details about using these functions, see [HYPERLINK](#) on page 148 and [IMAGE](#) on page 150.

Traffic Lights for Status

This formula displays a green, yellow, or red traffic light images to indicate status, using a custom picklist field called Project Status. Use this formula in list views and reports to create a “Status Summary” dashboard view.

```
IMAGE (
CASE(Project_Status__c,
"Green", "/img/samples/light_green.gif",
"Yellow", "/img/samples/light_yellow.gif",
"Red", "/img/samples/light_red.gif",
"/s.gif"),
"status color")
```

For details about using these functions, see [IMAGE](#) on page 150 and [CASE](#) on page 142.

Stars for Ratings

This formula displays a set of one to five stars to indicate a rating or score.

```
IMAGE (
CASE(Rating__c,
"1", "/img/samples/stars_100.gif",
"2", "/img/samples/stars_200.gif",
"3", "/img/samples/stars_300.gif",
"4", "/img/samples/stars_400.gif",
"5", "/img/samples/stars_500.gif",
"/img/samples/stars_000.gif"),
"rating")
```

For details about using these functions, see [IMAGE](#) on page 150 and [CASE](#) on page 142.

Consumer Reports™—Style Colored Circles for Ratings

This formula displays a colored circle to indicate a rating on a scale of one to five, where solid red is one, half red is two, black outline is three, half black is four, and solid black is five.

```
IMAGE (
CASE(Rating__c,
"1", "/img/samples/rating1.gif",
"2", "/img/samples/rating2.gif",
"3", "/img/samples/rating3.gif",
"4", "/img/samples/rating4.gif",
"5", "/img/samples/rating5.gif",
"/s.gif"),
"rating")
```

For details about using these functions, see [IMAGE](#) on page 150 and [CASE](#) on page 142.

Horizontal Bars to Indicate Scoring

This formula displays a horizontal color bar (green on a white background) of a length that is proportional to a numeric score. In this example, the maximum length of the bar is 200 pixels.

```
IMAGE("/img/samples/color_green.gif", "green", 15, Industry_Score__c * 2) &
IMAGE("/s.gif", "white", 15,
200 - (Industry_Score__c * 2))
```

For details about using this function, see [IMAGE](#) on page 150.

Integration Links

Application API Link

This formula creates a link to an application outside Database.com, passing the parameters so that it can connect to Database.com via the Web services API and create the necessary event.

```
HYPERLINK ("https://www.myintegration.com?sessionId=" & GETSESSIONID() & "?&rowId=" & Name &
"action=CreateTask","Create a Meeting Request")
```

For details about using these functions, see [HYPERLINK](#) on page 148 and [GETSESSIONID](#) on page 148.

Shipment Tracking Integration

This formula creates a link to FedEx, UPS, or DHL shipment tracking websites, depending on the value of a Shipping Method custom picklist field. Note that the parameters shown in this example for FedEx, UPS, and DHL websites are illustrative and do not represent the correct parameters for all situations.

```
CASE(Shipping_Method__c,
"Fedex",
HYPERLINK("http://www.fedex.com/Tracking?ascend_header=1&clienttype
=dotcom&cntry_code=us&language=english&tracknumbers= "& tracking_id__c,"Track"),
"UPS",
HYPERLINK("http://wwwapps.ups.com/WebTracking/processInputRequest?HTMLVersion
=5.0&sort_by=status&loc=en_US&InquiryNumber1= "& tracking_id__c & "&track.x=32&track.y=7",
"Track"),
"DHL",
HYPERLINK("http://track.dhl-usa.com/TrackByNbr.asp?ShipmentNumber=" & tracking_id__c,"Track"),
 "")
```

For details about using these functions, see [CASE](#) on page 142 and [HYPERLINK](#) on page 148.

Skype™ Auto Dialer Integration

This formula creates a linkable phone number field that automatically dials the phone number via the Skype VOIP phone application. It requires installation of the Skype application (a third-party product not provided by salesforce.com) on your desktop.

```
HYPERLINK("callto://+" & Country_Code__c & Phone_Unformatted__c, Phone)
```

For details about using this function, see [HYPERLINK](#) on page 148.

Metrics

Temperature Conversion

This formula converts Celsius degrees to Fahrenheit.

```
1.8 * degrees_celsius_c + 32
```

For details about using these operators, see [* \(Multiply\)](#) on page 137 and [+ \(Add\)](#) on page 136.

Unit of Measure Conversion

This formula converts kilometers to miles.

```
Miles_c / .621371192
```

For details about using this operator, see [/ \(Divide\)](#) on page 137.

Project Management

Calculate Intermediate Milestone from End Date

This formula calculates intermediate milestone dates by subtracting days from the end date (for projects that are planned based on end date).

```
Release_Date_c - 7 * Phase_duration_in_weeks_c
```

For details about using this operator, see [* \(Multiply\)](#) on page 137.

Operators and Functions

Use the following operators and functions when building formulas. Click on the name or description below to view more details. All functions are available everywhere that you can include a formula such as formula fields, validation rules, and workflow rules, unless otherwise specified.

 **Note:** Extraneous spaces in the samples below are ignored.

Math Operators

Operator	Description
+	Calculates the sum of two values.
-	Calculates the difference of two values.
*	Multiplies its values.
/	Divides its values.
^	Raises a number to a power of a specified number.
()	Specifies that the expressions within the open parenthesis and close parenthesis are evaluated first. All other expressions are evaluated using standard operator precedence.

Logical Operators

Operator	Description
= and ==	Evaluates if two values are equivalent.
<> and !=	Evaluates if two values are not equivalent.
<	Evaluates if a value is less than the value that follows this symbol.
>	Evaluates if a value is greater than the value that follows this symbol.
<=	Evaluates if a value is less than or equal to the value that follows this symbol.
>=	Evaluates if a value is greater than or equal to the value that follows this symbol.
&&	Evaluates if two values or expressions are both true. Use this operator as an alternative to the logical function AND.
	Evaluates if at least one of multiple values or expressions is true. Use this operator as an alternative to the logical function OR.

Text Operators

Operator	Description
&	Connects two or more strings.

Date and Time Functions

Function	Description
DATE	Returns a date value from year, month, and day values you enter. Database.com displays an error on the detail page if the value of the DATE function in a formula field is an invalid date, such as February 29 in a non-leap year.
DATEVALUE	Returns a date value for a date/time or text expression.
DATETIMEVALUE	Returns a year, month, day and GMT time value.
DAY	Returns a day of the month in the form of a number between 1 and 31.
MONTH	Returns the month, a number between 1 (January) and 12 (December) in number format of a given date.
NOW	Returns a date/time representing the current moment.
TODAY	Returns the current date as a date data type.
YEAR	Returns the four-digit year in number format of a given date.

Informational Functions

Function	Description
BLANKVALUE	Determines if an expression has a value and returns a substitute expression if it does not. If the expression has a value, returns the value of the expression.
ISBLANK	Determines if an expression has a value and returns TRUE if it does not. If it contains a value, this function returns FALSE.
ISNULL	Determines if an expression is null (blank) and returns TRUE if it is. If it contains a value, this function returns FALSE.
NULLVALUE	Determines if an expression is null (blank) and returns a substitute expression if it is. If the expression is not blank, returns the value of the expression.
PRIORVALUE	Returns the previous value of a field.

Logical Functions

Function	Description
AND	Returns a TRUE response if all values are true; returns a FALSE response if one or more values are false.
CASE	Checks a given expression against a series of values. If the expression is equal to a value, returns the corresponding result. If it is not equal to any values, it returns the <code>else_result</code> .
IF	Determines if expressions are true or false. Returns a given value if true and another value if false.
ISCHANGED	Compares the value of a field to the previous value and returns TRUE if the values are different. If the values are the same, this function returns FALSE.
ISNEW	Checks if the formula is running during the creation of a new record and returns TRUE if it is. If an existing record is being updated, this function returns FALSE.
ISNUMBER	Determines if a text value is a number and returns TRUE if it is. Otherwise, it returns FALSE.
NOT	Returns FALSE for TRUE and TRUE for FALSE.
OR	Determines if expressions are true or false. Returns TRUE if any expression is true. Returns FALSE if all expressions are false.

Math Functions

Function	Description
ABS	Calculates the absolute value of a number. The absolute value of a number is the number without its positive or negative sign.
CEILING	Rounds a number up to the nearest integer.
EXP	Returns a value for e raised to the power of a number you specify.

Function	Description
FLOOR	Returns a number rounded down to the nearest integer.
LN	Returns the natural logarithm of a specified number. Natural logarithms are based on the constant e value of 2.71828182845904.
LOG	Returns the base 10 logarithm of a number.
MAX	Returns the highest number from a list of numbers.
MIN	Returns the lowest number from a list of numbers.
MOD	Returns a remainder after a number is divided by a specified divisor.
ROUND	Returns the nearest number to a number you specify, constraining the new number by a specified number of digits.
SQRT	Returns the positive square root of a given number.

Text Functions

Function	Description
BEGINS	Determines if text begins with specific characters and returns TRUE if it does. Returns FALSE if it does not.
BR	Inserts a line break in a string of text.
CONTAINS	Compares two arguments of text and returns TRUE if the first argument contains the second argument. If not, returns FALSE.
FIND	Returns the position of a string within a string of text represented as a number.
GETSESSIONID	Returns the user's session ID.
HYPERLINK	Creates a link to a URL specified that is linkable from the text specified.
IMAGE	Inserts an image with alternate text and height/width specifications.
INCLUDES	Determines if any value selected in a multi-select picklist field equals a text literal you specify.
ISPICKVAL	Determines if the value of a picklist field is equal to a text literal you specify.
LEFT	Returns the specified number of characters from the beginning of a text string.
LEN	Returns the number of characters in a specified text string.
LOWER	Converts all letters in the specified text string to lowercase. Any characters that are not letters are unaffected by this function. Locale rules are applied if a locale is provided.
LPAD	Inserts characters you specify to the left-side of a text string.
MID	Returns the specified number of characters from the middle of a text string given the starting position.
RIGHT	Returns the specified number of characters from the end of a text string.

Function	Description
RPAD	Inserts characters that you specify to the right-side of a text string.
SUBSTITUTE	Substitutes new text for old text in a text string.
TEXT	Converts a percent, number, date, date/time, or currency type field into text anywhere formulas are used. Also, converts picklist values to text in validation rules, formula fields, and field updates.
TRIM	Removes the spaces and tabs from the beginning and end of a text string.
UPPER	Converts all letters in the specified text string to uppercase. Any characters that are not letters are unaffected by this function. Locale rules are applied if a locale is provided.
VALUE	Converts a text string to a number.

Summary Functions

The following functions can only be used in custom summary formulas for summary and matrix reports.

Function	Description
+ (Add)	
Description:	Calculates the sum of two values.
Use:	value1 + value2 and replace each <i>value</i> with merge fields, expressions, or other numeric values.
Formula Field Example:	<p>Amount + Maint_Amount__c + Services_Amount__c</p> <p>This formula calculates the sum of the product Amount, maintenance amount, and services fees. Note that Maint amount and Service Fees are custom currency fields.</p>
Validation Rule Example:	<p>You may have a custom object that allows users to track the total number of hours worked in a week. Use the following example to ensure that users cannot save a time card record with more than 40 hours in a work week.</p> <pre>Monday_Hours__c + Tuesday_Hours__c + Wednesday_Hours__c + Thursday_Hours__c + Friday_Hours__c > 40</pre> <p>Use a formula like this one in a validation rule to display the following error message when the total number of hours entered for each work day is greater than 40: "Your total hours cannot exceed 40." This example requires five custom fields on your custom object, one for each day of work.</p>

- (Subtract)

Description:	Calculates the difference of two values.
Use:	$value1 - value2$ and replace each <i>value</i> with merge fields, expressions, or other numeric values.
Example:	<p>Amount - Discount_Amount__c</p> <p>This formula calculates the difference of the product Amount less the Discount Amount. Note that Discount Amount is a custom currency field.</p>

*** (Multiply)**

Description:	Multiples its values.
Use:	$value1 * value2$ and replace each <i>value</i> with merge fields, expressions, or other numeric values.
Example:	<p>Consulting_Days__c * 1200</p> <p>This formula calculates the number of consulting days times 1200 given that this formula field is a currency data type and consulting charges a rate of \$1200 per day. Note that Consulting Days is a custom field.</p>

/ (Divide)

Description:	Divides its values.
Use:	$value1 / value2$ and replace each <i>value</i> with merge fields, expressions, or other numeric values.
Example:	<p>AnnualRevenue / NumberOfEmployees</p> <p>This formula calculates the revenue amount per employee using a currency field.</p>

^ (Exponentiation)

Description:	Raises a number to a power of a specified number.
Use:	$number^{integer}$ and replace <i>number</i> with a merge field, expression, or another numeric value; replace <i>integer</i> with a merge field that contains an integer, expression, or any integer.
Example:	NumberOfEmployees^4 calculates the number of employees to the 4th power.
Tips:	Avoid replacing <i>integer</i> with a negative number.

() (Open Parenthesis and Close Parenthesis)

Description:	Specifies that the expressions within the open parenthesis and close parenthesis are evaluated first. All other expressions are evaluated using standard operator precedence.
Use:	$(expression1) \ expression2\dots$ and replace each <i>expression</i> with merge fields, expressions, or other numeric values.
Example:	$(Unit_Value_c - Old_Value_c) / New_Value_c$ calculates the difference between the old value and new value divided by the new value.

= and == (Equal)

Description:	Evaluates if two values are equivalent.
Use:	$expression1=expression2$ or $expression1 == expression2$, and replace each <i>expression</i> with merge fields, expressions, or other numeric values.
Example:	Due Date Due Date = CreatedDate + 5 assigns a due date that is five days past the create date.

<> and != (Not Equal)

Description:	Evaluates if two values are not equivalent.
Use:	$expression1 <> expression2$ or $expression1 != expression2$, and replace each <i>expression</i> with merge fields, expressions, or other numeric values.
Example:	 <pre>IF(Maint_Amount_c + Services_Amount_c <> Amount, "DISCOUNTED", "FULL PRICE")</pre> This formula displays “DISCOUNTED” on product if its maintenance amount and services amount do not equal the product amount. Otherwise, displays “FULL PRICE.” Note that this example uses two custom currency fields for Maint Amount and Services Amount.

< (Less Than)

Description:	Evaluates if a value is less than the value that follows this symbol.
Use:	$value1 < value2$ and replace each <i>value</i> with merge fields, expressions, or other numeric values.

Example:

`IF(AnnualRevenue < 1000000, 1, 2)` assigns the value “1” with revenues less than one million and the value “2” to revenues greater than one million.

> (Greater Than)**Description:**

Evaluates if a value is greater than the value that follows this symbol.

Use:

`value1 > value2` and replace each `value` with merge fields, expressions, or other numeric values.

Example:

`IF(commission_c > 1000000, "High Net Worth", "General")` assigns the “High Net Worth” value to a commission greater than one million. Note, this is a text formula field that uses a commission custom field.

<= (Less Than or Equal)**Description:**

Evaluates if a value is less than or equal to the value that follows this symbol.

Use:

`value1 <= value2` and replace each `value` with merge fields, expressions, or other numeric values.

Example:

`IF(AnnualRevenue <= 1000000, 1, 2)` assigns the value “1” with revenues less than or equal to one million and the value “2” with revenues greater than one million.

>= (Greater Than or Equal)**Description:**

Evaluates if a value is greater than or equal to the value that follows this symbol.

Use:

`value1 >= value2` and replace each `value` with merge fields, expressions, or other numeric values.

Example:

`IF(Commission_c >= 1000000, "YES", "NO")` assigns the “YES” value with a commission greater than or equal to one million. Note, this is a text formula field that uses a custom currency field called `Commission`.

&& (AND)**Description:**

Evaluates if two values or expressions are both true. Use this operator as an alternative to the logical function AND.

Use:

`(logical1) && (logical2)` and replace `logical1` and `logical2` with the values or expressions that you want evaluated.

Example:

```
IF((Price<100 && Quantity<5),"Small", null)
```

This formula displays “Small” if the price is less than 100 and quantity is less than five. Otherwise, this field is blank.

|| (OR)**Description:**

Evaluates if at least one of multiple values or expressions is true. Use this operator as an alternative to the logical function OR.

Use:

`(logical1) || (logical2)` and replace any number of logical references with the values or expressions you want evaluated.

Validation Rule Example:

```
(Discount_Rate__c < 0) || (Discount_Rate__c > 0.40)
```

This validation rule formula displays the following error message when the `Discount Rate` custom field is not between 0 and 40%: "Discount Rate cannot exceed 40%."

& (Concatenate)**Description:**

Connects two or more strings.

Use:

`string1&string2` and replace each `string` with merge fields, expressions, or other values.

Example:

```
"Expense-" & Trip_Name__c & "-" & ExpenseNum__c
```

This formula displays the text “Expense-” followed by trip name and the expense number. This is a text formula field that uses an expense number custom field.

ABS**Description:**

Calculates the absolute value of a number. The absolute value of a number is the number without its positive or negative sign.

Use:

`ABS(number)` and replace `number` with a merge field, expression, or other numeric value that has the sign you want removed.

Example:

`ABS(ExpectedRevenue)` calculates the positive value of the `Expected Revenue` amount regardless of whether it is positive or negative.

AND

Description:	Returns a TRUE response if all values are true; returns a FALSE response if one or more values are false. Use this function as an alternative to the operator && (AND) .
Use:	AND(logical1, logical2, ...) and replace <i>logical1, logical2, ...</i> with the values that you want evaluated.
Formula Field Example:	<pre>IF(AND(Price<1,Quantity<1),"Small", null)</pre> <p>This formula displays “Small” if the price and quantity are less than one. This field is blank if the asset has a price or quantity greater than one.</p>

BEGINS

Description:	Determines if text begins with specific characters and returns TRUE if it does. Returns FALSE if it does not.
Use:	BEGINS (<i>text, compare_text</i>) and replace <i>text, compare_text</i> with the characters or fields you want to compare.
Example:	<pre>IF(BEGINS (Product_type__c, "ICU"), "Medical", "Technical")</pre> <p>This example returns the text “Medical” if the text in any Product Type custom text field begins with “ICU.” For all other products, it displays “Technical.”</p>
Tips:	<ul style="list-style-type: none"> This function is case sensitive so be sure your <i>compare_text</i> value has the correct capitalization. When using this function in a validation rule or workflow rule, fields that are blank are considered valid. For example, if you have a validation rule that tests to see if the serial number of an asset begins with “3,” all assets that have a blank serial number are considered valid.

BLANKVALUE

Description:	Determines if an expression has a value and returns a substitute expression if it does not. If the expression has a value, returns the value of the expression.
Use:	BLANKVALUE (<i>expression, substitute_expression</i>) and replace <i>expression</i> with the expression you want evaluated; replace <i>substitute_expression</i> with the value you want to replace any blank values.
Example:	<p>Example 1</p> <pre>BLANKVALUE(Department, "Undesignated")</pre>

This formula returns the value of the Department field if the Department field contains a value. If the Department field is empty, this formula returns the word Undesignated.

Example 2

```
(BLANKVALUE (Payment_Due_Date__c, StartDate +5)
```

This formula returns the date five days after the contract start date whenever Payment Due Date is blank. Payment Due Date is a custom date field.

Tips:

- Use BLANKVALUE instead of NULLVALUE in new formulas. BLANKVALUE has the same functionality as NULLVALUE, but also supports text fields. Database.com will continue to support NULLVALUE, so you do not need to change existing formulas.
- A field is not empty if it contains a character, blank space, or zero. For example, a field that contains a space inserted with the spacebar is not empty.
- Use the **BLANKVALUE** function to return a specified string if the field does not have a value; use the **ISBLANK** function if you only want to check if the field has a value.
- If you use this function with a numeric field, the function only returns the specified string if the field does not have a value and is not configured to treat blank fields as zeroes.

BR

Description:	Inserts a line break in a string of text.
Use:	BR ()
Tips:	<ul style="list-style-type: none"> • Do not remove the parentheses after the function name. • Keep the parentheses empty. They do not need to contain a value. • Remember to surround the BR() with concatenation operators: &.

CASE

Description:	Checks a given expression against a series of values. If the expression is equal to a value, returns the corresponding result. If it is not equal to any values, it returns the <code>else_result</code> .
Use:	<code>CASE (expression, value1, result1, value2, result2, ..., else_result)</code> and replace <code>expression</code> with the field or value you want compared to each specified value. Replace each value and result with the value that must be equivalent to return the result entry. Replace <code>else_result</code> with the value you want returned when the expression does not equal any values.
Default Value Example:	Product Language

You may want to associate a product with its language so that your users know the type of documentation or adapter to include. Use the following default value formula to automatically set the language of a product based on the country of the user creating the product. In this example, the default value is “Japanese” if the user’s country is “Japan” and “English” if the user’s country is “US.” If neither is true, the default value “unknown” is inserted into the Product Language field.

```
CASE($User.Country , "Japan", "Japanese",
"US", "English","unknown")
```

Tips:

- Be sure your *value1*, *value2*... expressions are the same data type.
- Be sure your *result1*, *result2*... expressions are the same data type.
- CASE functions cannot contain functions that return true or false. Instead, make true or false expressions return numbers such as:

```
CASE(1, IF(ISPICKVAL (Term_c, "12"), 1,
0),
12 * Monthly_Commit_c,
IF(ISPICKVAL(Term_c, "24"), 1, 0),
24 * Monthly_Commit_c, 0)
```

In this formula, Term is a picklist field that is multiplied by the Monthly Commit whenever it contains the value 1 for true.

- The *else_result* value is required.
- CASE functions return an error whenever any of the expressions return an error, regardless of which one should be returned.
- If the field in your CASE function is blank, it returns your *else_result* value. For example, this formula: CASE(Days_Open_c, 3, "Reassign", 2, "Assign Task", "Maintain") displays “Maintain” if the Days_Open field is blank, 0, or any value other than 2 or 3.
- Use CASE functions to determine if a picklist value is equal to a particular value. For example the formula CASE(Term_c, "12", 12 * Monthly_Commit_c, "24", 24 * Monthly_Commit_c, 0) multiplies the Monthly Commit amount by 12 whenever the Term is 12 or multiplies the Monthly Commit amount by 24 whenever the Term is 24. Otherwise, the result is zero.

CEILING

Description:	Rounds a number up to the nearest integer.
Use:	CEILING (<i>number</i>) and replace <i>number</i> with the field or expression you want rounded.
Example:	<p>Rounding Up (literal value)</p> <pre>CEILING(2.5)</pre> <p>This formula returns 3, which is 2.5 rounded up to the nearest number.</p> <p>Earthquake Magnitude</p> <p>CEILING (Magnitude_c) returns the value of a formula number field that calculates the magnitude of an earthquake up to the nearest integer.</p>

CONTAINS

Description:	Compares two arguments of text and returns TRUE if the first argument contains the second argument. If not, returns FALSE.
Use:	CONTAINS (<i>text</i> , <i>compare_text</i>) and replace <i>text</i> with the text that contains the value of <i>compare_text</i> .
Example:	<pre>IF(CONTAINS(Product_Type_c, "part"), "Parts", "Service")</pre> <p>This formula checks the content of a custom text field named Product_Type and returns “Parts” for any product with the word “part” in it. Otherwise, it returns “Service.” Note that the values are case sensitive, so if a Product_Type field contains the text “Part” or “PART,” this formula returns “Services.”</p>
Tips:	<ul style="list-style-type: none"> This function is case sensitive so be sure your <i>compare_text</i> value has the correct capitalization. When using this function in a validation rule or workflow rule, fields that are blank are considered valid. For example, if you have a validation rule that tests to see if the serial number of an asset contains “A,” all assets that have a blank serial number are considered valid. The CONTAINS function does not support multi-select picklists. Use INCLUDES to see if a multi-select picklist has a specific value.

DATE

Description:	Returns a date value from year, month, and day values you enter. Database.com displays an error on the detail page if the value of the DATE function in a formula field is an invalid date, such as February 29 in a non-leap year.
Use:	DATE (<i>year, month, day</i>) and replace <i>year</i> with a four-digit year, <i>month</i> with a two-digit month, and <i>day</i> with a two-digit day.
Example:	DATE (2005, 01, 02) creates a date field of January 2, 2005.

DATEVALUE

Description:	Returns a date value for a date/time or text expression.
Use:	DATEVALUE (<i>expression</i>) and replace <i>expression</i> with a date/time or text value, merge field, or expression.
Example:	<p>Closed Date DATEVALUE (ClosedDate) displays a date field based on the value of the Date/Time Closed field.</p> <p>Literal Date Value DATEVALUE ("2005-11-15") returns November 15, 2005 as a date value.</p>
Tips:	<ul style="list-style-type: none"> If the field referenced in the function is not a valid text or date/time field, the formula field displays #ERROR! When entering a date as a literal value, surround the date with quotes and use the following format: YYYY-MM-DD, that is, a four-digit year, two-digit month, and two-digit day. If the <i>expression</i> does not match valid date ranges, such as the MM is not between 01 and 12, the formula field displays #ERROR! Dates and times are always calculated using the user's time zone.

DATETIMEVALUE

Description:	Returns a year, month, day and GMT time value.
Use:	DATETIMEVALUE (<i>expression</i>) and replace <i>expression</i> with a date/time or text value, merge field, or expression.
Example:	<p>Closed Date DATETIMEVALUE (ClosedDate) displays a date field based on the value of the Date/Time Closed field.</p>

Literal Date Value

`DATETIMEVALUE ("2005-11-15 17:00:00")` returns November 15, 2005 5:00 PM GMT as a date and time value

Tips:

- `DATETIMEVALUE` is always calculated using GMT time zone and can't be changed.
- When entering a date as a literal value, surround the date with quotes and use the following format: YYYY-MM-DD, that is, a four-digit year, two-digit month, and two-digit day.
- If the *expression* does not match valid date ranges, such as the MM is not between 01 and 12, the formula field displays #ERROR!

DAY

Description:	Returns a day of the month in the form of a number between 1 and 31.
Use:	<code>DAY (date)</code> and replace <i>date</i> with a date field or value such as <code>TODAY()</code> .
Example:	<code>DAY (Code_Freeze_c)</code> returns the day in your custom code freeze date. Note this does not work on date/time fields.

EXP

Description:	Returns a value for e raised to the power of a number you specify.
Use:	<code>EXP (number)</code> and replace <i>number</i> with a number field or value such as 5.
Example:	<p>Exponent of a Literal Value</p> <p><code>EXP (3)</code></p> <p>This formula returns the value of e to the third power.</p> <p>Compound Interest</p> <p><code>Principal_c * EXP (Rate_c * Years_c)</code></p> <p>This formula calculates the compound interest based on a custom currency field for principal, custom percent field for rate, and custom number field for years.</p>

FIND

Description:	Returns the position of a string within a string of text represented as a number.
---------------------	---

Use:

`FIND(search_text, text[, start_num])` and replace *search_text* with the string you want to find, replace *text* with the field or expression you want to search, and replace *start_num* with the number of the character from which to start searching from left to right.

Example:**Street Address**

`FIND(" ", Street)` returns the character position of the first space in the *Street* field. You can use this number to find out the length of the street address as a means of separating a street address from street name in an address field.

Deriving Website Addresses

`SUBSTITUTE(Email, LEFT(Email, FIND("@", Email)), "www.")` finds the location of the @ sign in a person's email address to determine the length of text to replace with a "www." as a means of deriving their website address.

Tips:

- Be sure to remove the brackets, [and], from your formula before validating it.
- If the field referenced in your *text* parameter is blank, the formula field displays 0.
- Your *search_text* parameter is case sensitive and cannot contain any wildcard characters.
- If your search does not return any results, a 0 displays in the field.
- The *start_num* parameter is optional. If you do not enter a *start_num* value, the formula uses the value one, or the first character in the string.
- If your *start_num* is not greater than zero, a 0 displays in the field.
- If your *start_num* is greater than the length of the text, a 0 displays in the field.
- When entering your *start_num* parameter, remember that some fields like the *Website* field are unique because a "http://" is automatically appended to the beginning of the text you enter.
- Note that the first character in a string is designated as one rather than zero.

FLOOR**Description:**

Returns a number rounded down to the nearest integer.

Use:

`FLOOR(number)` and replace *number* with a number field or value such as 5.245.

Example:**Commission Amounts**

`FLOOR(commission__c)` rounds commission down to the nearest integer.

GETRECORDIDS**Description:**

Returns an array of strings in the form of record IDs for the selected records in a list, such as a list view or related list.

Use:

`{ !GETRECORDIDS (object_type) }` and replace *object_type* with a reference to the custom or standard object for the records you want to retrieve.

Tips:

- Use global variables to access special merge fields for s-controls, custom buttons, and links.
- Activities are special types of objects. Use `{!GETRECORDIDS($ObjectType.Task)}` when creating a task list button. Use `{!GETRECORDIDS($ObjectType.Event)}` when creating an event list button.
- This function is only available in custom buttons, links, and s-controls.

GETSESSIONID**Description:**

Returns the user's session ID.

Use:

`GETSESSIONID()`

Example:

```
HYPERLINK  
("https://www.myintegration.com?sId=" &  
GETSESSIONID() & "?&rowID=" & Name &  
"action=CreateTask", "Create  
a Meeting Request")
```

creates a link to an application outside of Database.com, passing the parameters so that it can connect to Database.com via the API and create the necessary event.

HYPERLINK**Description:**

Creates a link to a URL specified that is linkable from the text specified.

Use:

`HYPERLINK (url, friendly_name [, target])` and replace *url* with the Web address, replace *friendly_name* with the link text, and, optionally, replace *target* with the window or frame in which to display the content.

Tips:

- Hyperlink formula fields are of type text.
- Include the protocol and URL in quotes as in `Hyperlink("http://www.cnet.com", "cnet")`.
- Avoid using text functions such as LEN, LEFT, or RIGHT on HYPERLINK function results.
- When linking to Database.com pages, use a relative link, such as “`00U/e?retURL=%...`”, for hyperlink formulas.
- Use the `$Api` variable to reference API URLs.
- Be sure to remove the brackets, [and], from your formula before validating it.
- The `target` parameter is optional. If you do not specify a `target`, the link opens in a new browser window. Some common `target` parameters are:

_blank

Displays link in a new unnamed window.

_self

Displays link in the same frame or window as the element that refers to it.

_parent

Displays link in the immediate frameset parent of the current frame. This value is the same as `_self` if the current frame has no parent.

_top

Displays link in the full original window, canceling any other frames. This value is the same as `_self` if the current frame has no parent.

For more information on basic HTML tags, consult an HTML reference on the Internet.

- The HYPERLINK function is available everywhere that you can define a formula except [default values](#), [field updates](#), [validation rules](#), and [workflow rules](#).

IF**Description:**

Determines if expressions are true or false. Returns a given value if true and another value if false.

Use:

`IF(logical_test, value_if_true, value_if_false)` and replace `logical_test` with the expression you want evaluated; replace `value_if_true` with the value you want returned if the expression is true; replace `value_if_false` with the value you want returned if the expression is false.

Formula Field Example:**Overdue Payments**

```
IF(AND(Payment_Due_Date__c < TODAY(),
Payment_Status__c = "UNPAID") , "PAYMENT
OVERDUE", null)
```

This formula determines if the payment due date is past and the payment status is “UNPAID.” If so, returns the text “PAYMENT OVERDUE” and if not, leaves the field blank. This example uses a custom date field called Payment Due Date and a text custom field called Payment Status.

Insert Tax Rate

Use this default value formula to set the tax rate of an asset based on the user's city. Create a custom percent field with the following default value:

```
IF($User.City = "Napa", 0.0750,
IF($User.City = "Paso Robles", 0.0725,
IF($User.City = "Sutter Creek", 0.0725,
IF($User.City = "Los Olivos", 0.0750,
IF($User.City = "Livermore", 0.0875,
null
)
)
)
)
)
```

Tips:

- Make sure your *value_if_true* and *value_if_false* expressions are the same data type.

IMAGE**Description:**

Inserts an image with alternate text and height/width specifications.

Use:

`IMAGE(image_url, alternate_text, height, width)` and replace *image_url* with the full path to the image; replace *alternate_text* with the string of text you want displayed when you hover your mouse over the image; replace *height* with the vertical size of the image in pixels; replace *width* with the horizontal size of the image in pixels.

Example:

```
HYPERLINK("ymsgr:sendIM?" & Yahoo_Name__c,
IMAGE("http://opi.yahoo.com/online?u=" &
Yahoo_Name__c & "&m=g&t;=0", "Yahoo"))
```

This formula displays a clickable Yahoo! Messenger icon indicating if the person is logged on to the service. Users can click the icon to launch a Yahoo! Messenger conversation with the person. This example uses a custom text field called Yahoo Name on contacts where you can store the contact's Yahoo! Messenger ID.

Tips:

- The *height* and *width* parameters are optional.
- Use a text string to replace the *image_url* and *alternate_text* parameters. Surround each text string in quotes.
- Use numbers to replace the *height* and *width* parameters.
- If you use Internet Explorer, you may need to change your security settings so that it does not display a warning prompt when images use HTTP protocol. See the online help for Internet Explorer for instructions on changing your security settings.
- The IMAGE function cannot include the GETSESSIONID function as one of its arguments.
- The IMAGE function is available everywhere that you can define a formula except [default values](#), [field updates](#), [validation rules](#), and [workflow rules](#).

INCLUDES

Description:	Determines if any value selected in a multi-select picklist field equals a text literal you specify.
Use:	<code>INCLUDES (multiselect_picklist_field, text_literal)</code> and replace <i>multiselect_picklist_field</i> with the merge field name for the multi-select picklist; and replace <i>text_literal</i> with the multi-select picklist value you want to match in quotes.
Examples:	<code>INCLUDES (Hobbies__c, "Golf")</code> returns TRUE if one of the selected values in the Hobbies custom multi-select picklist field is Golf.
Tips:	<ul style="list-style-type: none"> • The <i>text_literal</i> expression must be of type text and enclosed in quotes. It cannot be a merge field or the result of a function. • Database.com returns an error if any of the following occurs: <ul style="list-style-type: none"> ◊ You do not provide a <i>text_literal</i> expression. ◊ You provide an empty <i>text_literal</i> expression, such as "" or " ". • Use ISNULL to determine if a multi-select picklist field is empty. • Use the PRIORVALUE function inside the INCLUDES function to check if the previous value of a multi-select picklist field included a specific value. For example: <pre>INCLUDES (PRIORVALUE (multiselect_picklist_field), text_literal)</pre>

ISBLANK

Description:	Determines if an expression has a value and returns TRUE if it does not. If it contains a value, this function returns FALSE.
Use:	ISBLANK (<i>expression</i>) and replace <i>expression</i> with the expression you want evaluated.
Example:	<pre>(IF(ISBLANK(Maint_Amount_c), 0, 1) + IF(ISBLANK(Services_Amount_c), 0, 1) + IF(ISBLANK(Discount_Percent_c), 0, 1) + IF(ISBLANK(Amount), 0, 1) + IF(ISBLANK(Timeline_c), 0, 1)) / 5</pre> <p>This formula takes a group of fields and calculates what percent of them are being used by your personnel. This formula field checks five fields to see if they are blank. If so, a zero is counted for that field. A “1” is counted for any field that contains a value and this total is divided by five (the number of fields evaluated). Note that this formula requires you select the Treat blank fields as blanks option under Blank Field Handling while the Advanced Formula subtab is showing.</p>
Tips:	<ul style="list-style-type: none"> • Use ISBLANK instead of ISNULL in new formulas. ISBLANK has the same functionality as ISNULL, but also supports text fields. Database.com will continue to support ISNULL, so you do not need to change any existing formulas. • A field is not empty if it contains a character, blank space, or zero. For example, a field that contains a space inserted with the spacebar is not empty. • Use the BLANKVALUE function to return a specified string if the field does not have a value; use the ISBLANK function if you only want to check if the field has a value. • If you use this function with a numeric field, the function only returns TRUE if the field has no value and is not configured to treat blank fields as zeroes.

ISCHANGED

Description:	Compares the value of a field to the previous value and returns TRUE if the values are different. If the values are the same, this function returns FALSE.
Use:	ISCHANGED (<i>field</i>) and replace <i>field</i> with the name of the field you want to compare.

Validation Rule Example:

The following validation rule prevents users from changing an object name after it has been created:

```
NOT (ISCHANGED (Name) ).
```

`NOT (AND (ISCHANGED (Priority), ISPICKVAL (Priority, "Low")))` is a validation rule that ensures if a user changes the Priority of a case, the new priority cannot be “Low.”

`NOT (AND (ISCHANGED (CloseDate), OR (MONTH (CloseDate) <> MONTH (TODAY ()), YEAR (CloseDate) <> YEAR (TODAY ())), $Profile.Name <> "Sales Manager"))` is a validation rule that prevents a user from changing the Close Date of an opportunity to a date outside of the current month and year unless that user has the “Sales Manager” profile.



Note: \$Profile merge fields are only available in Enterprise, Unlimited, and Developer Editions.

Tips:

- This function is available only in:
 - ◊ Assignment rules
 - ◊ Validation rules
 - ◊ Field updates
 - ◊ Workflow rules if the trigger type is set to Every time a record is created or edited.
- Use the NOT function to reverse the return values of TRUE and FALSE.
- This function returns FALSE when evaluating any field on a newly created record.
- If a text field was previously blank, this function returns TRUE when it contains any value.
- For number, percent, or currency fields, this function returns TRUE when:
 - ◊ The field was blank and now contains any value
 - ◊ The field was zero and now is blank
 - ◊ The field was zero and now contains any other value

ISNEW**Description:**

Checks if the formula is running during the creation of a new record and returns TRUE if it is. If an existing record is being updated, this function returns FALSE.

Use:

```
ISNEW ()
```

Validation Rule Example:

Use the following validation rule to prevent users from creating a record with a close date in the past. AND (ISNEW(), CloseDate < TODAY()) checks if the user is creating a new opportunity and, if so, ensures that the Close Date is today or after today.

Use this validation rule to ensure users add at least one product to an opportunity after they have created it.

```
NOT(OR(ISNEW(), HasOpportunityLineItem))
```

In this example, the validation rule formula displays the following error message when an existing opportunity does not have any products: "You must add products to this opportunity before saving." This does not display an error on the initial save because they cannot add products until after saving the record initially; but it prevents them from resaving or closing an opportunity that does not contain products.

Tips:

- This function is available only in validation rules, field updates, and workflow rules.
- Use the NOT function to reverse the return values of TRUE and FALSE.
- This function always returns FALSE when used in a workflow rule with a time-based trigger.
- This function always returns FALSE when used in a field update for an approval action.

ISNULL**Description:**

Determines if an expression is null (blank) and returns TRUE if it is. If it contains a value, this function returns FALSE.



Note: Use ISBLANK instead of ISNULL in new formulas. ISBLANK has the same functionality as ISNULL, but also supports text fields. Database.com will continue to support ISNULL, so you do not need to change any existing formulas.

Use:

ISNULL(*expression*) and replace *expression* with the expression you want evaluated.

Example:

```
(IF(ISNULL(Maint_Amount__c), 0, 1) +
 IF(ISNULL(Services_Amount__c), 0, 1) +
 IF(ISNULL(Discount_Percent__c), 0, 1) +
 IF(ISNULL(Amount), 0, 1) +
 IF(ISNULL(Timeline__c), 0, 1)) / 5
```

This formula takes a group of fields and calculates what percent of them are being used by your personnel. This formula field checks five fields to see if they are blank. If so, a zero is counted for that field. A “1” is counted for any field that contains a value and this total is divided by five (the number of fields evaluated). Note that this formula requires you select the Treat blank fields as blanks option under Blank Field Handling while the Advanced Formula subtab is showing.

Validation Rule Example:

```
AND(ISPICKVAL(StageName, "Closed Won"),
 ISNULL(Project_Start_Date__c))
```

This validation rule makes the Project Start Date custom date field conditionally required whenever the stage is “Closed Won.”

Tips:

- Text fields are never null, so using this function with a text field always returns false. For example, the formula field `IF(ISNULL(new__c) 1, 0)` is always zero regardless of the value in the New field. For text fields, use the `ISBLANK` function instead.
- Empty date and date/time fields always return true when referenced in `ISNULL` functions.
- Choose Treat blank fields as blanks for your formula when referencing a number, percent, or currency field in an `ISNULL` function. Choosing Treat blank fields as zeroes gives blank fields the value of zero so none of them will be null.
- Merge fields can be handled as blanks, which can affect the results of components that call this function.
- When using a validation rule to ensure that a number field contains a specific value, use the `ISNULL` function to include fields that do not contain any value. For example, to validate that a custom field contains a value of '1,' use the following validation rule to display an error if the field is blank or any other number:

```
OR(ISNULL(field__c), field__c<>1)
```

ISNUMBER

Description:	Determines if a text value is a number and returns TRUE if it is. Otherwise, it returns FALSE.
Use:	ISNUMBER (<i>text</i>) and replace <i>text</i> with the merge field name for the text field.
Validation Rule Example:	<pre>OR(LEN(Bank_Account_Number__c) <> 10, NOT(ISNUMBER(Bank_Account_Number__c)))</pre> <p>This validation rule ensures a custom text field called Bank Account Number is a number of 10 digits and is not blank.</p>
Tips:	<ul style="list-style-type: none"> This function returns FALSE for blank values. The ISNUMBER function is not aware of your locale. For example, ISNUMBER ("123,12") and ISNUMBER ("1 000") return FALSE even if the user's locale is "French." Chinese, Japanese, Korean, and special characters including a space return FALSE. The ISNUMBER function returns TRUE for scientific formatting such as "2E2" or "123.123."

ISPICKVAL

Description:	Determines if the value of a picklist field is equal to a text literal you specify.
Use:	ISPICKVAL (<i>picklist_field</i> , <i>text_literal</i>) and replace <i>picklist_field</i> with the merge field name for the picklist; replace <i>text_literal</i> with the picklist value in quotes. <i>text_literal</i> cannot be a merge field or the result of a function.
Tips:	<ul style="list-style-type: none"> Replace <i>picklist_field</i> with a custom field of type picklist. Your <i>text_literal</i> expression must be of type text and enclosed in quotes. It cannot be a merge field or the result of a function. Use CASE functions to determine if a picklist value is equal to a particular value. When using the ISPICKVAL function to return the previous value of a picklist field, include the PRIORVALUE function inside the ISPICKVAL function as in this example: <pre>ISPICKVAL(PRIORVALUE (<i>picklist_field</i>), <i>text_literal</i>)</pre>

JSENCODE

Description:	Encodes text and merge field values for use in JavaScript by inserting escape characters, such as a backslash (\), before unsafe JavaScript characters, such as the apostrophe (').
Use:	{ !JSENCODE (text) } and replace text with the merge field or text string that contains the unsafe JavaScript characters.
Example:	If the merge field foo_c contains Enter the user's name, { !JSENCODE (foo_c) } results in: \u003CB\u003EEnter the user\'s name\u003C\;/b\u003E

JSINHTMLENCODE

Description:	Encodes text and merge field values for use in JavaScript within HTML tags by inserting escape characters before unsafe JavaScript characters and replacing characters that are reserved in HTML with HTML entity equivalents.
Use:	{ !JSINHTMLENCODE (text) } and replace text with the merge field or text string that contains the unsafe JavaScript characters.
Example:	If the merge field foo_c contains Enter the user's name, { !JSINHTMLENCODE (foo_c) } results in: Enter the user's name

LEFT

Description:	Returns the specified number of characters from the beginning of a text string.
Use:	LEFT (text , num_chars) and replace text with the field or expression you want returned; replace num_chars with the number of characters from the left you want returned.
Example:	TRIM(LEFT(LastName, 5)) & "-" & TRIM(RIGHT(SSN_c, 4)) This formula displays the first five characters of a name and the last four characters of a social security number separated by a dash. Note that this example uses a text custom field called SSN.
Tips:	<ul style="list-style-type: none"> Reference auto-number fields as text fields in formulas. If the num_chars value is less than zero, Database.com replaces the value with zero.

LEN

Description:	Returns the number of characters in a specified text string.
Use:	<code>LEN (text)</code> and replace <i>text</i> with the field or expression whose length you want returned.
Example:	<code>LEN (PartNumber_c)</code> This formula returns the number of characters in a Product Code field.

LN

Description:	Returns the natural logarithm of a specified number. Natural logarithms are based on the constant e value of 2.71828182845904.
Use:	<code>LN (number)</code> and replace <i>number</i> with the field or expression for which you want the natural logarithm. Note: the LN function is the inverse of the EXP function.
Example:	<code>LN (10)</code> returns the natural logarithm of 10, which is 2.30. <code>LN (Value_c)</code> returns the natural logarithm of a custom number field called Value.

LOG

Description:	Returns the base 10 logarithm of a number.
Use:	<code>LOG (number)</code> and replace <i>number</i> with the field or expression from which you want the base 10 logarithm calculated.
Example:	Salary <code>LOG (Salary_c)</code> calculates the logarithm of a person's salary. In this example, Salary is a custom currency field. Hydrogen <code>-LOG (Hydrogen_c)</code> calculates the pH and acidity using the LOG function and a custom number field called Hydrogen, which represents the concentration of Hydrogen ions in the liquid measured in moles per liter.

LOWER

Description:	Converts all letters in the specified text string to lowercase. Any characters that are not letters are unaffected by this function. Locale rules are applied if a locale is provided.
Use:	<code>LOWER (text, [locale])</code> and replace <i>text</i> with the field or text you wish to convert to lowercase, and <i>locale</i> with the

<p>Example:</p>	<p>MYCOMPANY.COM</p> <p>LOWER ("MYCOMPANY.COM") returns "mycompany.com."</p> <p>Ticker Symbol</p> <p>LOWER(TickerSymbol) returns the text in Ticker Symbol in lower case characters.</p> <p>Applying Turkish Language Locale Rules</p> <p>The Turkish language has two versions of the letter i: one dotted and one dotless. The locale rules for Turkish require the ability to capitalize the dotted i, and allow the dotless I to be lowercase. To correctly use the LOWER() function with the Turkish language locale, use the Turkish locale code <i>tr</i> in the LOWER() function as follows:</p> <pre>LOWER(<i>text</i>, "tr")</pre> <p>This ensures that Database.com does not transform any dotted i in the <i>text</i> to a dotless I.</p>
------------------------	--

LPAD

<p>Description:</p>	<p>Inserts characters you specify to the left-side of a text string.</p>
<p>Use:</p>	<p>LPAD(<i>text</i>, <i>padded_length</i>[, <i>pad_string</i>]) and replace the variables:</p>
	<ul style="list-style-type: none"> • <i>text</i> is the field or expression you want to insert characters to the left of. • <i>padded_length</i> is the number of total characters in the text that will be returned. • <i>pad_string</i> is the character or characters that should be inserted. <i>pad_string</i> is optional and defaults to a blank space.
	<p>If the value in <i>text</i> is longer than <i>pad_string</i>, <i>text</i> is truncated to the size of <i>padded_length</i>.</p>
<p>Example:</p>	<p>Field Name: Padding</p> <p>LPAD(Name, 20) truncates the Name field after 20 characters. For example, if the name is mycompany.com, the value returned is "mycompany.com."</p> <p>My_Company: No Change</p> <p>LPAD('my_company.com', 14, 'z') returns "my_company.com" without change because it has 14 characters.</p> <p>Field Name Padded with Z</p>

`LPAD(Name, 15, 'z')` returns the name "zmycompany.com."

Field Name: Truncating

`LPAD(Name, 2)` truncates the name after the second character. For example, if the name is `mycompany.com`, the value returned is "my."

Tips:

Leading blank spaces and zeros are omitted.

MAX

Description:	Returns the highest number from a list of numbers.
Use:	<code>MAX(number, number, ...)</code> and replace <code>number</code> with the fields or expressions from which you want to retrieve the highest number.
Example:	<p>Service Charge</p> <pre>MAX(0.06 * Total_Cost__c, Min_Service_Charge__c)</pre> <p>In this example, the formula field calculates a service charge of 6% of the total cost or a minimum service charge, whichever is greater. Note that <code>Min Service Charge</code> is a custom currency field with a default value of \$15. However, you could make it a formula field if your minimum service charge is always the same amount.</p> <p>This formula determines which amount to pay in royalties for a book. It displays the greater of two amounts: \$0.07 for each book sold or \$0.10 per page. It assumes you have custom number fields for <code>Pages</code> and <code>Total Sold</code> and a custom currency field for <code>Retail Price</code>.</p> <p>Commissions</p> <pre>MAX(\$User.Commission_Percent__c * Price, Price * Account_Discount__c, 100)</pre> <p>This formula determines what commission to log for an asset based on which is greater: the user's commission percentage of the price, the price times the discount percent stored for the account or 100 dollars. This example assumes you have two custom percent fields on users and assets.</p>

MID

Description:	Returns the specified number of characters from the middle of a text string given the starting position.
---------------------	--

Use:

`MID(text, start_num, num_chars)` and replace *text* with the field or expression to use when returning characters; replace *start_num* with the number of characters from the left to use as a starting position; replace *num_chars* with the total number of characters to return.

MIN**Description:**

Returns the lowest number from a list of numbers.

Use:

`MIN(number, number, ...)` and replace *number* with the fields or expressions from which you want to retrieve the lowest number.

Example:**401K Matching**

```
MIN(250, Contribution_c / 2)
```

This example formula determines which amount to provide in employee 401K matching based on a matching program of half of the employee's contribution or \$250, whichever is less. It assumes you have custom currency field for Contribution.

Bonus

```
MIN(Gross_c * Bonus_Percent_c,
     Performance_c / Number_of_Employees_c)
```

This example determines an employee's bonus amount based on the smallest of two amounts: the employee's gross times bonus percent or an equally divided amount of the company's performance amount among all employees. It assumes you have custom number field for Number of Employees, a custom percent field for Bonus Percent, and currency custom fields for the employee's Gross and company's Performance.

MOD**Description:**

Returns a remainder after a number is divided by a specified divisor.

Use:

`MOD(number, divisor)` and replace *number* with the field or expression you want divided; replace *divisor* with the number to use as the divisor.

Example:

`MOD(3, 3)` returns 0

`MOD(4, 3)` returns 1

`MOD(123, 100)` returns 23

You may want to prevent users from scheduling meetings on a Saturday or Sunday. Use the following example to apply a validation rule to a custom date field called My Date.

```
CASE (MOD(My_Date__c - DATE(1900, 1, 7), 7),
0, 0,
6, 0,
1) = 0
```

This example displays the following error message when the value of My Date is not Monday through Friday: "My Date is not a weekday."

MONTH

Description:	Returns the month, a number between 1 (January) and 12 (December) in number format of a given date.
Use:	MONTH (<i>date</i>) and replace <i>date</i> with the field or expression for the date containing the month you want returned.
Example:	<p>SLA Expiration</p> <p>MONTH (SLAExpirationDate__c) returns the month that your service-level agreement expires. This example uses a custom date field called SLA Expiration Date.</p> <p>Current Month</p> <p>MONTH (TODAY()) returns the current month in a number format. For example, the month of February would be the value "2."</p>

NOT

Description:	Returns FALSE for TRUE and TRUE for FALSE.
Use:	NOT (<i>logical</i>) and replace <i>logical</i> with the expression that you want evaluated.
Example:	IF(NOT (ISPICKVAL(Status, "Closed")), ROUND(NOW() - CreatedDate, 0), null) checks to see if a variable is open and if so, calculates the number of days it has been open by subtracting the date and time created from the current date and time. The result is the number of days open rounded to zero decimal places. If the variable is not open, this field is blank.

NOW

Description:	Returns a date/time representing the current moment.
Use:	NOW()
Tips:	<ul style="list-style-type: none"> • Do not remove the parentheses. • Keep the parentheses empty. They do not need to contain a value. • Use a date/time field in a NOW function instead of a date field. Created Date and Last Modified Date are date/time fields whereas Last Activity Date is a date field. • Use TODAY if you prefer to use a date field.

- Dates and times are always calculated using the user's time zone.
- Use addition and subtraction operators with a NOW function and other date/time fields to return a number, representing number of days. For example `NOW() - CreatedDate` calculates the number of days since the created date of a record. In this example, the formula field data type is a number.
- Use addition and subtraction operators with a NOW function and numbers to return a date and time. For example `NOW() + 5` calculates the date and time five days ahead of now. In this example, the formula field data type is a date/time.

NULLVALUE

Description:	Determines if an expression is null (blank) and returns a substitute expression if it is. If the expression is not blank, returns the value of the expression.
	<p> Note: Use <code>BLANKVALUE</code> instead of <code>NULLVALUE</code> in new formulas. <code>BLANKVALUE</code> has the same functionality as <code>NULLVALUE</code>, but also supports text fields. Database.com will continue to support <code>NULLVALUE</code>, so you do not need to change existing formulas.</p>
Use:	<code>NULLVALUE(<i>expression</i>, <i>substitute_expression</i>)</code> and replace <i>expression</i> with the expression you want to evaluate; replace <i>substitute_expression</i> with the value you want to replace any blank values.
Example:	<pre>(NULLVALUE(Sample_Due_Date__c, StartDate +5))</pre> <p>This formula returns the date five days after the start date whenever Sample Due Date is blank. Sample Due Date is a custom date field.</p>
Tips:	<ul style="list-style-type: none"> • Avoid using this function with text fields because they are never null even when they are blank. Instead, use the <code>BLANKVALUE</code> function to determine if a text field is blank. • Choose Treat blank fields as blanks for your formula when referencing a number, percent, or currency field in a <code>NULLVALUE</code> function. Choosing Treat blank fields as zeroes gives blank fields the value of zero so none of them will be null. • Use the same data type for both the <i>expression</i> and <i>substitute_expression</i>.

OR

Description:	Determines if expressions are true or false. Returns TRUE if any expression is true. Returns FALSE if all expressions are false. Use this function as an alternative to the operator <code> (OR)</code> .
Use:	<code>OR(<i>logical1</i>, <i>logical2</i>...)</code> and replace any number of logical references with the expressions you want evaluated.
Formula Field Example:	<pre>IF(OR(ISPICKVAL(Priority, "High"), ISPICKVAL(Status, "New")), ROUND(NOW()-CreatedDate, 0), null)</pre> <p>This formula returns the number of days a case has been open if the Status is new or the Priority is high. If the case was opened today, this field displays a zero.</p>

Validation Rule Example:

```
OR(Sample_Rate__c < 0, Sample_Rate__c > 0.40)
```

This validation rule formula displays the following error message when the Sample Rate custom field is not between 0 and 40%: “SampleRate cannot exceed 40%.”

PRIORVALUE

Description:	Returns the previous value of a field.
Use:	<code>PRIORVALUE (<i>field</i>)</code>
Tips:	<ul style="list-style-type: none"> This function is available only in: <ul style="list-style-type: none"> Assignment rules Validation rules Field updates Workflow rules if the trigger type is set to Every time a record is created or edited. This function does not return default values. When users create a new record, this function returns the value of the <i>field</i> referenced rather than null. For example, if you create an account named “Acme,” <code>PRIORVALUE (Account.Name)</code> returns Acme. When using the ISPICKVAL function to return the previous value of a picklist field, include the PRIORVALUE function inside the ISPICKVAL function as in this example: <pre>ISPICKVAL(PRIORVALUE (<i>picklist_field</i>), <i>text_literal</i>)</pre> <ul style="list-style-type: none"> Use the PRIORVALUE function inside the INCLUDES function to check if the previous value of a multi-select picklist field included a specific value. For example: <pre>INCLUDES (PRIORVALUE(<i>multiselect_picklist_field</i>), <i>text_literal</i>)</pre>

REGEX

Description:	Compares a text field to a regular expression and returns TRUE if there is a match. Otherwise, it returns FALSE. A regular expression is a string used to describe a format of a string according to certain syntax rules.
Use:	<code>REGEX (<i>text</i>, <i>regex_text</i>)</code> and replace <i>text</i> with the text field, and <i>regex_text</i> with the regular expression you want to match.

Validation Rule Example:

This example ensures that a custom field called SSN matches a regular expression representing a valid social security number format of the form 999-99-9999.

```
NOT (
    OR (
        LEN (SSN__c) = 0,
        REGEX(SSN__c, "[0-9]{3}-[0-9]{2}-[0-9]{4}")
    )
)
```

Tips:

- Regular expression syntax is based on [Java Platform SE 6 syntax](#). However, backslash characters (\) must be changed to double backslashes (\\\) because backslash is an escape character in Database.com.
- The Database.com regular expression engine matches an entire string as opposed to searching for a match within a string. For example, if you are searching for the name Marc Benioff, use the regular expression, .*Marc Benioff.*, to find a match in a string like the following:

According to Marc Benioff, the social enterprise increases customer success.

If you use the regular expression, Marc Benioff, the only string that this regular expression will match is:

Marc Benioff

- Capture groups and substitutions are ignored.
- This function is available everywhere formulas exist except formula fields and default values.

REQUIRESCRIPT

Description: Returns a script tag with source for a URL you specify. Use this function when referencing the Force.com AJAX Toolkit or other JavaScript toolkits.

Use: { !REQUIRESCRIPT (*url*) } and replace *url* with the link for the script that is required.

For the AJAX Toolkit:

```
{!requireScript("/soap/ajax/13.0/connection.js")}
```

Returns:

```
<script src="/soap/ajax/13.0/connection.js"></script>
```

For Dojo:

```
{!requireScript("/js/dojo/0.3.1/dojo.js")}
```

Returns:

```
<script src="/js/dojo/0.3.1/dojo.js"></script>
```

Tips:

- Use global variables to access special merge fields for s-controls.
- Use this function when creating custom buttons or links where you have set the Behavior to “Execute JavaScript” and Content Source to “OnClick JavaScript” because the script tag should be outside the OnClick code.
- This function is only available for custom buttons and links that have Content Source set to “OnClick JavaScript.”
- When working in Visualforce, use INCLUDESCRIPT instead.

RIGHT

Description:	Returns the specified number of characters from the end of a text string.
Use:	RIGHT (<i>text</i> , <i>num_chars</i>) and replace <i>text</i> with the field or expression you want returned; replace <i>num_chars</i> with the number of characters from the right you want returned.
Example:	TRIM(LEFT(LastName, 5)) & "-" & TRIM(RIGHT(SSN_c, 4)) displays the first five characters of the a name and the last four characters of asocial security number separated by a dash. Note that this assumes you have a text custom field called SSN.
Tips:	<ul style="list-style-type: none"> • Reference auto-number fields as text fields in formulas. • If the <i>num_chars</i> value is less than zero, Database.com replaces the value with zero.

ROUND

Description:	Returns the nearest number to a number you specify, constraining the new number by a specified number of digits.
Use:	ROUND (<i>number</i> , <i>num_digits</i>) and replace <i>number</i> with the field or expression you want rounded; replace <i>num_digits</i> with the number of decimal places you want to consider when rounding.
Example:	<pre>ROUND (1.5, 0) = 2 ROUND (1.2345, 0) = 1 ROUND (-1.5, 0) = -2 ROUND (225.49823, 2) = 255.50</pre>
Tips:	<ul style="list-style-type: none"> • Enter zero for <i>num_digits</i> to round a number to the nearest integer. • Database.com automatically rounds numbers based on the decimal places you specify. For example, a custom number field with two decimal places stores 1.50 when you enter 1.49999. • Database.com uses the round half-up rounding algorithm. Half-way values are always rounded up. For example, 1.45 is rounded to 1.5. -1.45 is rounded to -1.5.

- The decimal numbers displayed depend on the decimal places you selected when defining the field in the custom field wizard. The *num_digits* represents the number of digits considered when rounding.

RPAD

Description:	Inserts characters that you specify to the right-side of a text string.
Use:	<p><code>RPAD(<i>text</i>, <i>padded_length</i>[, '<i>pad_string</i>')]</code> and replace the variables:</p> <ul style="list-style-type: none"> <i>text</i> is the field or expression after which you want to insert characters. <i>padded_length</i> is the number of total characters in the text string that will be returned. <i>pad_string</i> is the character or characters that should be inserted. <i>pad_string</i> is optional and defaults to a blank space. <p>If the value in <i>text</i> is longer than <i>pad_string</i>, <i>text</i> is truncated to the size of <i>padded_length</i>.</p>
Example:	<p>Field Name: Padding Default</p> <p><code>RPAD(Name, 20)</code> truncates the Name field after 20 characters. For example, if the name is <code>mycompany.com</code>, the value returned is "mycompany.com."</p> <p>My_Company: No Change</p> <p><code>RPAD('my_company.com', 14, 'z')</code> returns "my_company.com" without change because it has 14 characters.</p> <p>Field Name: Padding with a Character</p> <p><code>RPAD(Name, 15, 'z')</code> returns "mycompany.comz".</p> <p>Field Name: Truncating</p> <p><code>RPAD(Name, 2)</code> truncates the name after the second character. For example, if the name is <code>mycompany.com</code>, the value returned is "my."</p>
Tips:	Ending blank spaces are omitted.

SQRT

Description:	Returns the positive square root of a given number.
Use:	<code>SQRT(<i>number</i>)</code> and replace <i>number</i> with the field or expression you want computed into a square root.
Example:	<p><code>SQRT(25)</code> returns the square root of 25, which is 5.</p> <p>Amplitude</p> <p><code>SQRT(Amplitude_c)</code> returns the square root of a custom number field representing the amplitude of an earthquake.</p>
Tips:	<ul style="list-style-type: none"> Calculating the square root of a negative number results in an error on the detail page.

- Avoid division by zero errors by including an IF function such as:
`IF(Amplitude_c >= 0, SQRT(Amplitude_c), null).`

SUBSTITUTE

Description:	Substitutes new text for old text in a text string.
Use:	<code>SUBSTITUTE(text, old_text, new_text)</code> and replace <i>text</i> with the field or value for which you want to substitute values, <i>old_text</i> with the text you want replaced, and <i>new_text</i> with the text you want to replace the <i>old_text</i> .
Example:	<code>SUBSTITUTE>Email, LEFT>Email, FIND("@", Email)), "www."</code> finds the location of the @ sign in a person's email address to determine the length of text to replace with a "www." as a means of deriving their website address.
Tips:	<ul style="list-style-type: none"> Each term provided in quotes is case sensitive. If the <i>old_text</i> appears more than once, each occurrence is replaced with the <i>new_text</i> value provided even when that results in duplicates.

TEXT

Description:	Converts a percent, number, date, date/time, or currency type field into text anywhere formulas are used. Also, converts picklist values to text in validation rules, formula fields, and field updates.
Use:	<code>TEXT(value)</code> and replace <i>value</i> with the field or expression you want to convert to text format. Avoid using any special characters besides a decimal point (period) or minus sign (dash) in this function.
Example:	<p>Asset ID</p> <p><code>SerialNumber &"-"& TEXT(Quantity)</code> returns an asset ID number starting with the serial number and ending with the quantity separated by a dash. The Serial Number field is already text but the Quantity field is a number, requiring the TEXT function before it.</p> <p>Use Picklist Values in Math Equations</p> <pre>VALUE(LEFT(TEXT(Quantity), 5)) * Unit</pre> <p>This formula multiplies the first five numbers of the Quantity picklist by the Unit numeric field.</p>
Validation Rule Examples:	<p>Use Numeric Functions on Numeric Picklist Values</p> <p><code>VALUE(LEFT(TEXT(Quantity), 5)) * Unit > 10000</code> multiplies the first five numbers of the Quantity picklist by the Unit numeric field, and returns TRUE if the result is greater than 10,000.</p> <p>Directly Compare Two Picklists</p>

`TEXT(bug_status) = TEXT(case_status)` compares the values of the bug_status picklist with values of the case_status picklist, and returns TRUE if they are equal.

Tips:

- The returned text is not formatted with any currency, percent symbols, or commas.
- Values are not sensitive to locale. For example, 24.42 EUR are converted into the number 24.42.
- Percents are returned in the form of a decimal.
- Dates are returned in the form of YYYY-MM-DD, that is, a four-digit year and two-digit month and day.
- Date/time values are returned in the form of YYYY-MM-DD HH:MM:SSZ where YYYY is a four-digit year, MM is a two-digit month, DD is a two-digit day, HH is the two-digit hour, MM are the minutes, SS are the seconds, and Z represents the zero meridian indicating the time is returned in UTC time zone.
- Picklist fields are only supported in TEXT functions used in validation rule formulas, formula fields, and field updates. In other formulas, use ISPICKVAL or CASE when referencing a picklist field.
- The TEXT function always returns picklist values in your organization's master language, not the language of the current user.

TODAY

Description:	Returns the current date as a date data type.
Use:	<code>TODAY()</code>
Example:	<code>TODAY() - Sample_date_c</code> calculates how many days in the sample are left.
Validation Rule Example:	<pre>SampleDate < TODAY()</pre> <p>This example ensures that users cannot change the Sample Date to any date in the past.</p>
Tips:	<ul style="list-style-type: none"> Do not remove the parentheses. Keep the parentheses empty. They do not need to contain a value. Use a date field with a TODAY function instead of a date/time field. Last Activity Date is a date field whereas Created Date and Last Modified Date are date/time fields. See NOW if you prefer to use a date/time field. Dates and times are always calculated using the user's time zone. Use addition and subtraction operators with a TODAY function and other date fields to return a number, representing number of days. For example <code>TODAY() - LastActivityDate</code> calculates the number of days since the last activity date. In this example, the formula field data type is a number. Use addition and subtraction operators with a TODAY function and numbers to return a date. For example <code>TODAY() + 5</code> calculates the date five days ahead of today. In this example, the formula field data type is a date.

TRIM

Description:	Removes the spaces and tabs from the beginning and end of a text string.
Use:	<code>TRIM(text)</code> and replace <code>text</code> with the field or expression you want to trim.
Example:	<code>TRIM(LEFT(LastName,5)) & "-" & RIGHT(FirstName, 1)</code> returns a network ID for users that contains the first five characters of their last name and first character of their first name separated by a dash.

UPPER

Description:	Converts all letters in the specified text string to uppercase. Any characters that are not letters are unaffected by this function. Locale rules are applied if a locale is provided.
Use:	<code>UPPER(text, [locale])</code> and replace <code>text</code> with the field or expression you wish to convert to uppercase, and <code>locale</code> with the optional two-character ISO language code or five-character locale code, if available..
Example:	<p>MYCOMPANY.COM</p> <p><code>UPPER("mycompany.com")</code> returns "MYCOMPANY.COM."</p> <p>MYCOMPANY.COM 123</p> <p><code>UPPER("Mycompany.com 123")</code> returns "MYCOMPANY.COM 123."</p> <p>Applying Turkish Language Locale Rules</p> <p>The Turkish language has two versions of the letter i: one dotted and one dotless. The locale rules for Turkish require the ability to capitalize the dotted i, and allow the dotless I to be lowercase. To correctly use the <code>UPPER()</code> function with the Turkish language locale, use the Turkish locale code <code>tr</code> in the <code>UPPER()</code> function as follows:</p> <p><code>UPPER(text, "tr")</code></p> <p>This ensures that any dotted i in the <code>text</code> does not transform to a dotless I.</p>

URLENCODE

Description:	Encodes text and merge field values for use in URLs by replacing characters that are illegal in URLs, such as blank spaces, with the code that represent those characters as defined in <i>RFC 3986, Uniform Resource Identifier (URI): Generic Syntax</i> . For example, blank spaces are replaced with %20, and exclamation points are replaced with %21.
Use:	<code>{!URLENCODE(text)}</code> and replace <code>text</code> with the merge field or text string that you want to encode.
Example:	If the merge field <code>foo_c</code> contains <code>Mark's page</code> , <code>{!URLENCODE(foo_c)}</code> results in: <code>%3CB%3EMark%27s%20page%3C%2FB%3E</code>
Tips:	<ul style="list-style-type: none"> Custom buttons and links with URL content sources have separate encoding settings. If you use the <code>URLENCODE</code> function to encode a custom button or link that has an

encoding setting specified, Database.com first encodes the URL according to the custom button or link setting, then encodes the result. For example, if the URL in a custom link contains a space and its encoding setting is UTF8, Database.com first encodes the space to a plus sign (+), then the URLENCODE function converts the plus sign to its character code, %2B.

- When you include the standard Account field on opportunities (Opportunity.Account) in the URLENCODE function, the value of the field is the account ID, not the account name. To encode the account name, create a custom cross-object formula field on opportunities that spans to the account name, and use that field in the URLENCODE function instead of Opportunity.Account. For example, if the cross-object formula is AccountNameFormula__c, use the following:

```
http://www.google.com/search?q={!URLENCODE(Opportunity.AccountNameFormula__c)}
```

VALUE

Description:	Converts a text string to a number.
Use:	VALUE (<i>text</i>) and replace <i>text</i> with the field or expression you want converted into a number.
Tips:	<p>Make sure the text in a VALUE function does not include any special characters other than a decimal point (period) or minus sign (dash). For example, the formula VALUE (Text_field__c) produces these results:</p> <ul style="list-style-type: none"> If Text field is 123, the result is 123 If Text field is blank, the result is #Error! If Text field is \$123, the result is #Error! If Text field is EUR123, the result is #Error!

VLOOKUP

Description:	Returns a value by looking up a related value on a custom object similar to the VLOOKUP() Excel function.
Use:	VLOOKUP (<i>field_to_return</i> , <i>field_on_lookup_object</i> , <i>lookup_value</i>) and replace <i>field_to_return</i> with the field that contains the value you want returned, <i>field_on_lookup_object</i> with the field on the related object that contains the value you want to match, and <i>lookup_value</i> with the value you want to match.
Validation Rule Example:	This example checks that a billing postal code is valid by looking up the first five characters of the value in a custom object called Zip_Code__c that contains a record for every valid zip code in the US. If the zip code is not found in the Zip_Code__c

object or the billing state does not match the corresponding State_Code_c in the Zip_Code_c object, an error is displayed.

```

AND (
LEN(BillingPostalCode) > 0,
OR(BillingCountry = "USA",
    BillingCountry = "US"),
VLOOKUP(
$ObjectType.Zip_Code_c.Fields.State_Code_c,
$ObjectType.Zip_Code_c.Fields.Name,
LEFT(BillingPostalCode,5))

<> BillingState
)

```



Note:

- Use this example when the billing country is US or USA.
- You can download US zip codes in CSV file format from <http://zips.sourceforge.net>.

Tips:

- The *field_to_return* must be an auto number, roll-up summary, lookup relationship, master-detail relationship, checkbox, date, date/time, email, number, percent, phone, picklist, text, text area, or URL field type.
- The *field_on_lookup_object* must be the Record Name field on a custom object.
- The *field_on_lookup_object* and *lookup_value* must be the same data type.
- If more than one record matches, the value from the first record is returned.
- The value returned must be on a custom object.
- You cannot delete the custom field or custom object referenced in this function.
- This function is only available in [validation rules](#).

YEAR

Description:	Returns the four-digit year in number format of a given date.
Use:	YEAR(date) and replace <i>date</i> with the field or expression that contains the year you want returned.
Example:	YEAR(TODAY()) - YEAR(Initial_Meeting_c) returns the number of years since your initial meeting with a client. This example uses a custom date field called Initial Meeting.

Managing Validation Rules

About Validation Rules

Improve the quality of your data using validation rules. Validation rules verify that the data a user enters in a record meets the standards you specify before the user can save the record. A validation rule can contain a formula or expression that evaluates the data in one or more fields and returns a value of “True” or “False.” Validation rules also include an error message to display to the user when the rule returns a value of “True” due to an invalid value.

After you have defined validation rules:

1. The user chooses to create a new record or edit an existing record.
2. The user clicks **Save**.
3. All validation rules are verified.
 - If all data is valid, the record is saved.
 - If any data is invalid, the associated error message displays without saving the record.
4. The user makes the necessary changes and clicks **Save** again.

You can specify the error message to display when a record fails validation and where to display it. For example, your error message can be “The close date must occur after today’s date.” You can choose to display it near a field or at the top of the page. Like all other error messages, validation rule errors display in red text and are preceded by the word “Error.”



Important: Validation rules apply to all new and updated records for an object, even if the fields referenced in the validation rule are not included in an API call. If your organization has any integrations that use this object, verify that the validation rule functions as intended for each integration.

Validation Rule Considerations

Validation rules verify that the data a user enters in a record meets the standards you specify before the user can save the record. A validation rule can contain a formula or expression that evaluates the data in one or more fields and returns a value of “True” or “False.” Validation rules also include an error message to display to the user when the rule returns a value of “True” due to an invalid value. Review the following implementation notes and best practices before implementing validation rules in your organization.

Implementation Notes

- It isn’t necessary to begin your validation rule formula with an IF function. Any Boolean error condition expression works. For example:
 - ◊ Correct: `CloseDate < TODAY()`
 - ◊ Incorrect: `IF(CloseDate < TODAY(), TRUE, FALSE)`
- Validation rules and lookup filters achieve similar ends, but offer different advantages. Use a lookup filter if:
 - Validation formulas can’t reference merge fields for auto number or compound address fields like `Mailing Address`. However, merge fields for individual address fields, such as `Billing City`, can be used in validation formulas.
 - Validation rules can’t refer to compound fields. Examples of compound fields include addresses, first and last names, dependent picklists, and dependent lookups..
 - The Data Loader and the Force.com API version 7 and later run validation rules.
 - Database.com runs validation rules on records before they are imported. Records that fail validation aren’t imported. Consider deactivating the appropriate validation rules before running an import if they affect the records you are importing.
 - Because updates to records based on workflow rules don’t trigger validation rules, workflow rules can invalidate previously valid fields.
 - An “Invalid Formula” error displays if your formula has a run-time error such as division by zero.
 - You can’t create validation rules for relationship group members.
 - When a validation rule contains the **BEGINS** or **CONTAINS** functions, it processes blank fields as valid. For example, if you have a validation rule that tests to see if the serial number of an asset begins with “3,” all assets that have a blank serial number are considered valid.
 - Validation rules don’t run on multiple records updated after a change owner or mass transfer. However, changing an owner of a single record does run validation rules.

Best Practices

- When creating validation rules, consider all the settings in your organization that can make a record fail validation such as [field updates](#) or [field-level security](#).
- Be careful not to create two contradicting validation rules for the same field; otherwise, users won't be able to save the record.
- A poorly designed validation rule can prevent users from saving valid data. Make sure you thoroughly test a validation rule before activating it. Users will never be able to save a record if your formula always returns a "True" value.
- Write helpful error messages:

Always include the field label.

Users may not know what field is failing validation, especially if your error message is located at the top of the page.

Give instructions.

An error message like "invalid entry" doesn't tell them what type of entry is valid. Use an error message like "Close Date must be after today."

Assign error numbers to validation rules and error messages.

This allows you to identify the source of the error.

- Use the record type ID merge field in your formula to apply different validation for different record types. For information about merge fields, see [Merge Fields Overview](#) on page 109.
- When using a validation rule to ensure that a number field contains a specific value, use the ISNULL function to include fields that do not contain any value. For example, to validate that a custom field contains a value of '1,' use the following validation rule to display an error if the field is blank or any other number:

```
OR(ISNULL(field__c), field__c<>1)
```

- When referencing related fields in your validation formula, make sure those objects are deployed.
- Check the [Debug Log](#) to monitor details of the start and completion of each validation rule evaluated.
- You can still validate the values of encrypted fields using [validation rules](#) or Apex. Both work regardless of whether the user has the "View Encrypted Data" permission. Data for encrypted fields in the debug log is masked.

Managing Validation Rules

User Permissions Needed	
To view field validation rules:	"View Setup and Configuration"
To define or change field validation rules:	"Customize Application"

Validation rules verify that the data a user enters in a record meets the standards you specify before the user can save the record. A validation rule can contain a formula or expression that evaluates the data in one or more fields and returns a value of "True" or "False." Validation rules also include an error message to display to the user when the rule returns a value of "True" due to an invalid value.

To begin using validation rules, click **Customize > Users**, select the appropriate activity, and click **Validation Rules**. For custom objects, click **Create > Objects** and select the custom object. Validation rules are listed in the Validation Rules related list.

- To define a validation rule, click **New**. See [Defining Validation Rules](#) on page 175.
- To make changes to a validation rule, click **Edit**.

- To delete a validation rule, click **Del**.
- To view details about a validation rule, click field validation name.
- To clone a validation rule, select the rule you want to clone and click **Clone**.
- To activate a validation rule, click **Edit** next to the rule you want to activate, select **Active**, and click **Save**. Deselect **Active** to deactivate the rule.

Defining Validation Rules

User Permissions Needed	
To view field validation rules:	“View Setup and Configuration”
To define or change field validation rules:	“Customize Application”

Validation rules verify that the data a user enters in a record meets the standards you specify before the user can save the record. A validation rule can contain a formula or expression that evaluates the data in one or more fields and returns a value of “True” or “False.” Validation rules also include an error message to display to the user when the rule returns a value of “True” due to an invalid value.

Before creating validation rules, review the [Validation Rule Considerations](#) on page 173.

To create validation rules:

1. For custom objects, click **Create > Objects** and select the custom object.
2. Click **New** in the Validation Rules related list.
3. Enter the properties of your validation rule:

Field	Description
Rule Name	Unique identifier of up to 80 characters with no spaces or special characters such as extended characters.
Active	Checkbox that indicates if the rule is enabled.
Description	A 255 character or less description that distinguishes the validation rule from others. For internal purposes only.
Error Condition Formula	The expression used to validate the field. See Building Formulas on page 118 and Operators and Functions on page 132.
Error Message	The message that displays to the user when a field fails the validation rule.

4. Click **Check Syntax** to check your formula for errors.
5. Click **Save** to finish or **Save & New** to create additional validation rules.

Examples of Validation Rules

User Permissions Needed	
To view field validation rules:	"View Setup and Configuration"
To define or change field validation rules:	"Customize Application"

Use the following samples for validation rules in Database.com and Force.com AppExchange apps, including:

- [User, Role, and Profile Validation Rules](#)
- [Date Validation Rules](#)
- [Number Validation Rules](#)
- [Other Validation Rules](#)

Number Validation Rules

Time Cards Must Total 40 Hours

Field	Value
Description:	Ensures that users cannot save a time card record with more than 40 hours in a work week. This example requires five custom fields on your custom object, one for each day of work.
Formula:	<pre>Monday_Hours__c + Tuesday_Hours__c + Wednesday_Hours__c + Thursday_Hours__c + Friday_Hours__c > 40</pre>
Error Message:	Your total hours cannot exceed 40.
Error Location:	Top of Page

Number Cannot Be Negative

Field	Value
Description:	Validates that a custom field called Hours Worked is not a negative number.
Formula:	<pre>Hours_Worked__c < 0</pre>
Error Message:	Hours Worked cannot be less than zero.
Error Location:	Hours Worked

Number Must Be Even

Field	Value
Description:	Validates that a custom field called Ark Passengers is a non-negative even number.

Field	Value
Formula:	<pre>OR(Ark_Passengers__c < 0, MOD(Ark_Passengers__c, 2) <> 0)</pre>
Error Message:	Ark Passengers must be a positive even number.
Error Location:	Ark Passengers

Number Must Be Odd

Field	Value
Description:	Validates that a custom field called Socks Found is a non-negative odd number.
Formula:	<pre>OR(Socks_Found__c < 0, MOD(Socks_Found__c, 2) = 0)</pre>
Error Message:	Socks Found must be an odd number.
Error Location:	Socks Found

Number Must Be a Multiple of Five

Field	Value
Description:	Validates that a custom field called Multiple of 5 is a multiple of five.
Formula:	<pre>MOD(Multiple_of_5__c, 5) <> 0</pre>
Error Message:	Number must be a multiple of five.
Error Location:	Multiple of 5

Number Must Be an Integer

Field	Value
Description:	Validates that a custom field called My Integer is an integer.
Formula:	<pre>FLOOR(My_Integer__c) <> My_Integer__c</pre>
Error Message:	This field must be an integer.
Error Location:	My Integer

Number Must Be Between -50 and 50

Field	Value
Description:	Validates that a custom field called <code>Volume</code> is between -50 and 50.
Formula:	$\text{ABS}(\text{Volume_c}) > 50$
Error Message:	Volume must be between -50 and 50.
Error Location:	Volume

Number Range Validation

Field	Value
Description:	Validates that the range between two custom fields, <code>Salary Min</code> and <code>Salary Max</code> , is no greater than \$20,000.
Formula:	$(\text{Salary_Max_c} - \text{Salary_Min_c}) > 20000$
Error Message:	Salary range must be within \$20,000. Adjust the Salary Max or Salary Min values.
Error Location:	Salary Max

Percentage Must Be Between Zero and 100

Field	Value
Description:	Validates that a custom field called <code>Mix Pct</code> is between 0 and 100%. Note that percent fields are expressed divided by 100 in formulas (100% is expressed as 1; 50% is expressed as 0.5).
Formula:	OR($\text{Mix_Pct_c} > 1.0,$ $\text{Mix_Pct_c} < 0.0$)
Error Message:	Mix Pct must be between 0 and 100%.
Error Location:	Mix Pct

Date Validation Rules

Date Must Be a Weekday

Field	Value
Description:	Validates that the value of a custom date field is a weekday (not Saturday or Sunday).

Field	Value
Formula:	<pre>CASE (MOD(My_Date__c - DATE(1900, 1, 7), 7), 0, 0, 6, 0, 1) = 0</pre>
Error Message:	Date must be a weekday.
Error Location:	My Date

Date Must Be a Weekend Day

Field	Value
Description:	Validates that the value of a custom date field is a Saturday or Sunday.
Formula:	<pre>CASE(MOD(My_Date__c - DATE(1900, 1, 7), 7), 0, 1, 6, 1, 0) = 0</pre>
Error Message:	Date must be a weekend day.
Error Location:	My Date

Date Must Be in the Current Month

Field	Value
Description:	Validates that a custom date field contains a date within the current month and year.
Formula:	<pre>OR (YEAR(My_Date__c) <> YEAR(TODAY()), MONTH(My_Date__c) <> MONTH(TODAY()))</pre>
Error Message:	Date must be in the current month.
Error Location:	My Date

Date Must Be in the Current Year

Field	Value
Description:	Validates that a custom date field contains a date within the current year.
Formula:	<pre>YEAR(My_Date__c) <> YEAR(TODAY())</pre>
Error Message:	Date must be in the current year.

Field	Value
Error Location:	My Date

Date Must Be the Last Day of the Month

Field	Value
Description:	Validates whether a custom field called My Date is the last day of the month. To do this, it determines the date of the first day of the next month and then subtracts 1 day. It includes special case logic for December.
Formula:	$\text{DAY}(\text{My_Date_c}) <> \\ \text{IF}(\text{Month}(\text{My_Date_c})=12, 31, \\ \text{DAY}(\text{DATE}(\text{YEAR}(\text{My_Date_c}), \text{MONTH}(\text{My_Date_c})+1, 1) - 1))$
Error Message:	Date must be the last day of the month.
Error Location:	My Date

Date Must Be Within One Year of Today

Field	Value
Description:	Validates whether a custom field called Follow-Up Date is within one year of today's date. This example assumes a 365 day year. (It does not handle leap years.)
Formula:	$\text{Followup_Date_c} - \text{TODAY}() > 365$
Error Message:	Follow-Up Date must be within one year of today.
Error Location:	Follow-Up Date

Day of Month Cannot Be Greater Than 15

Field	Value
Description:	Validates that a custom field called Begin Date contains a date in the first 15 days of the specified month.
Formula:	$\text{DAY}(\text{Begin_Date_c}) > 15$
Error Message:	Begin Date cannot be after the 15th day of month.
Error Location:	Begin Date

End Date Cannot Be Before Begin Date

Field	Value
Description:	Validates that a custom field called End Date does not come before another custom field called Begin Date.
Formula:	<code>Begin_Date__c > End_Date__c</code>
Error Message:	End Date cannot be before Begin Date.
Error Location:	Begin Date

Expiration Date Cannot Be Before Close Date

Field	Value
Description:	Validates that a custom field called Expiration Date does not come before Close Date.
Formula:	<code>Expiration_Date__c < CloseDate</code>
Error Message:	Expiration Date cannot be before Close Date.
Error Location:	Expiration Date

User, Role, and Profile Validation Rules

Expense Amount Does Not Exceed User's Max Allowed Expense

Field	Value
Description:	Validates a custom field called Expense Amount against a custom user field called Max Allowed Expense.
Formula:	<code>Expense_Amount__c > \$User.Max_Allowed_Expense__c</code>
Error Message:	Amount cannot exceed your maximum allowed expense.
Error Location:	Expense Amount

Only Record Owner Can Change Field

Field	Value
Description:	Ensures that only the record owner can make changes to a custom field called Personal Goal.

Field	Value
Formula:	<pre>AND (ISCHANGED(Personal_Goal__c), Owner <> \$User.Id)</pre>
Error Message:	Only record owner can change Personal Goal.
Error Location:	Personal Goal

Only Record Owner or Administrator Can Change Field

Field	Value
Description:	Ensures that a user can make changes to a custom field called Personal Goal only if the user is the record owner or has a custom profile of “Custom: System Admin.”
Formula:	<pre>AND (ISCHANGED(Personal_Goal__c), Owner <> \$User.Id, \$Profile.Name <> "Custom: System Admin")</pre>
	 Note: \$Profile merge fields are only available in Enterprise, Unlimited, and Developer Editions.
Error Message:	Only record owner or administrator can change Personal Goal.
Error Location:	Personal Goal

Other Validation Rules

Allow Number to Be Increased but Not Decreased

Field	Value
Description:	Allows a custom field called Commit Amount to be increased but not decreased after initial creation. This rule uses the PRIORVALUE() function to compare the updated value of the field to its value prior to update.
Formula:	PRIORVALUE(Commit_Amount__c) > Commit_Amount__c
Error Message:	Commit Amount cannot be decreased.
Error Location:	Commit Amount

California Driver's License

Field	Value
Description:	Ensures that a custom field called Drivers_License is in the correct A9999999 format when the Mailing_State is "CA".
Formula:	<pre>AND(MailingState = "CA", NOT(REGEX(Drivers_License__c, "([A-Z]\\\d{7})?")))</pre>
Error Message:	Invalid California driver's license format.
Error Location:	Drivers License

Force Users to Check “I Accept Terms” to Enter Certain Values

Field	Value
Description:	Uses a checkbox labeled “I accept terms” to force the user to select a checkbox in order to enter a value called Number_of_Days that exceeds their Paid Time Off (PTO) balance available.
Formula:	<pre>AND(NOT(I_accept_terms__c), Number_of_Days__c > \$User.PTO_Balance__c)</pre>
Error Message:	Request will cause a negative PTO balance. You must accept Negative PTO Balance terms.
Error Location:	I accept terms

Prohibit Changes to a Field After It Has Been Saved

Field	Value
Description:	Prevents users from changing a custom field called Guaranteed_Rate after it has been saved initially.
Formula:	<pre>AND(NOT(ISNEW()), ISCHANGED(Guaranteed_Rate__c))</pre>
Error Message:	Guaranteed Rate cannot be changed.
Error Location:	Guaranteed Rate

Social Security Number Format

Field	Value
Description:	Validates that a custom text field called SSN is formatted in 999-99-9999 number format (if it is not blank). The pattern specifies: <ul style="list-style-type: none"> Three single digits (0-9):\d{3} A dash Two single digits (0-9):\d{2} A dash Four single digits (0-9):\d{4}
Formula:	<pre>NOT (OR (ISBLANK(Social_Security_Number__c), REGEX(Social_Security_Number__c , "[0-9]{3}-[0-9]{2}-[0-9]{4}")))</pre>
Error Message:	SSN must be in this format: 999-99-9999.
Error Location:	SSN

Valid Currency

Field	Value
Description:	Validates selected currency against an explicit subset of active currencies in your organization using the Currency picklist. Use this example if you only allow some of the active currencies in your organization to be applied to certain types of records.
Formula:	<pre>CASE (CurrencyIsoCode, "USD", 1, "EUR", 1, "GBP", 1, "JPY", 1, 0) = 0</pre>
Error Message:	Currency must be USD, EUR, GBP, or JPY.
Error Location:	Currency

Valid Credit Card Number

Field	Value
Description:	Validates that a custom text field called Credit_Card_Number is formatted in 9999-9999-9999-9999 or 9999999999999999 number format when it is not blank. The pattern specifies: <ul style="list-style-type: none"> Four digits (0-9) followed by a dash: \d{4}- The aforementioned pattern is repeated three times by wrapping it in () {3}

Field	Value
	<ul style="list-style-type: none"> Four digits (0-9) The OR character () allows an alternative pattern of 16 digits of zero through nine with no dashes: \d{16}
Formula:	<pre>NOT(REGEX(Credit_Card_Number_c , "((\\d{4}-){3}\\d{4}) \\d{16})?"))</pre>
Error Message:	Credit Card Number must be in this format: 9999-9999-9999-9999 or 9999999999999999.
Error Location:	Credit Card Number

Valid IP Address

Field	Value
Description:	Ensures that a custom field called IP Address is in the correct format, four 3-digit numbers (0-255) separated by periods.
Formula:	<pre>NOT(REGEX(IP_Address_c , "^((25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?)\\.)\{3\} (25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?)\\$"))</pre>
Error Message:	Error: IP Address must be in form 999.999.999.999 where each part is between 0 and 255.
Error Location:	IP Address

Website Extension

Field	Value
Description:	Validates a custom field called Web Site to ensure its last four characters are in an explicit set of valid website extensions.
Formula:	<pre>AND(RIGHT(Web_Site_c, 4) <> ".COM", RIGHT(Web_Site_c, 4) <> ".com", RIGHT(Web_Site_c, 4) <> ".ORG", RIGHT(Web_Site_c, 4) <> ".org", RIGHT(Web_Site_c, 4) <> ".NET", RIGHT(Web_Site_c, 4) <> ".net", RIGHT(Web_Site_c, 6) <> ".CO.UK", RIGHT(Web_Site_c, 6) <> ".co.uk")</pre>
Error Message:	Web Site must have an extension of .com, .org, .net, or .co.uk.

Field	Value
Error Location:	Web Site

Managing Picklists

Picklist Considerations

User Permissions Needed
To change picklists: "Customize Application"

Customized selection lists, or “picklists,” let users pick values from a predefined list of entries. You can update the entries in picklists; see [Updating Picklists](#) on page 186.

Sorting Picklists

User Permissions Needed
To sort picklists: "Customize Application"

You can sort the values of picklist fields.

1. Navigate to the appropriate object:
 - Click **Create > Objects**, and click the name of the custom object in the list.
2. Click **Edit** next to the picklist you want to update.
3. Click **Reorder**.
4. Use the arrows to arrange the field in the proper sequence.
5. Select a default, if desired.
6. Check the box to alphabetize the entries for users on edit pages. The entries will always appear in alphabetical order, regardless of the user’s language.
7. Click **Save**.

Updating Picklists

User Permissions Needed
To change picklists: "Customize Application"

To update any picklist field:

1. Navigate to the fields page for your object. .

For custom objects, click **Create > Objects**, then click the name of an object.
2. In the Custom Fields & Relationships related list, click the name of the picklist field you want to update.
3. In the Picklist Values section, click **Edit** next to a value.
4. In the Picklist Edit page, you can change the name of the value and make the value the default for the master picklist.

**Tip:**

- Click **Save**.

Replacing Picklist Values

User Permissions Needed	
To replace picklist values:	“Customize Application”

You may need to replace a picklist value with another. For example, your status picklist has five values (Open, In Progress, Closed-red, Closed-yellow, and Closed-green) and you want to simplify it to three values (Open, In Progress, and Closed). You need to replace the Closed-red, Closed-yellow, and Closed-green values with a new value: Closed.



Note: Replacing an existing picklist value also changes the `Modified By` date and time for the record.

To globally replace the values of picklist fields in existing records follow the steps below.

- If necessary, create the replacement value in the picklist edit page, see [Updating Picklists](#) on page 186.
- Navigate to the appropriate object:
 - Click **Create > Objects**, and click the name of the custom object in the list.
- Click **Replace** next to the picklist.
- Type the exact value you want to change, and select a new replacement value.
- Check **Replace all blank values** to apply the new value on any records that do not have a value in this field.
- Click **Replace** to update all occurrences of the value with the new value in your organization's records including those in the Recycle Bin.
- In the Replace Picklist Confirmation page, click **Finished**.

Picklist Limitations

User Permissions Needed	
To change picklists:	“Customize Application”

The maximum number of entries you can have in a picklist is determined by the total number of characters allowed in the picklist, which is 15,000 characters. Note that each entry includes a linebreak and a return character that are not visible. These two additional characters per entry are counted as part of the 15,000 character limit.

Additional Limits for Custom Picklists

Within the 15,000 total character limit, custom picklists can have:

- Up to 1,000 entries
- Up to 255 characters per entry

Custom multi-select picklists can have:

- Up to 150 values
- Up to 40 characters per value

Note that for multi-select picklists, users can select up to 100 values at a time on a record.

Managing Dependent Picklists

About Dependent Picklists

Use dependent picklists to help your users enter accurate and consistent data. A *dependent picklist* is a custom or multi-select picklist for which the valid values depend on the value of another field, called the *controlling field*. Controlling fields can be any picklist (with at least one and fewer than 300 values) or checkbox field on the same record.

Defining Dependent Picklists

User Permissions Needed	
To define dependent picklists:	“Customize Application”

To define a dependent picklist:

1. Navigate to the appropriate object:
 - Click **Create > Objects**, and click the name of the custom object in the list.
2. Click **Field Dependencies**.
3. Click **New**.
4. Choose a controlling field and dependent field.



Note: Some picklist and checkbox fields may not be available as controlling fields. For a complete list of these fields, see [Dependent Picklist Considerations](#) on page 189.

5. Click **Continue**.
6. Use the **field dependency matrix** to specify the dependent picklist values that are available when a user selects each controlling field value.
7. Optionally, click **Preview** to test your selections.
8. Click **Save**.

Using the Field Dependency Matrix

User Permissions Needed	
To define picklist dependencies:	“Customize Application”

The field dependency matrix lets you specify the dependent picklist values that are available when a user selects each controlling field value. The top row of the matrix contains the controlling field values, while the columns list the dependent field values.

Use this matrix to include or exclude values. Included values are available in the dependent picklist when a value in the controlling field is selected. Excluded fields are not available in the dependent picklist for the selected controlling field value.

To include or exclude values:

- Double-click values to include them. Included values are indicated with highlighting. Double-click any highlighted values to exclude them.
- Click a value and use SHIFT+click on another value to select a range of adjacent values. Then click **Include Values** to make the values available, or **Exclude Values** to remove them from the list of available values.
- Click a value and use CTRL+click to select multiple values. Then click **Include Values** to make the values available, or **Exclude Values** to remove them from the list of available values.
- Click a column header to select all the values in that column. Then click **Include Values** to make the values available, or **Exclude Values** to remove them from the list of available values.

To change the values in your view:

- Click **View All** to view all available values at once.
- Click **Go To** and choose a controlling value to view all the dependent values in that column.
- Click **Previous** or **Next** to view the values in columns that are on the previous or next page.
- Click **View sets of 5** to view 5 columns at a time.

Dependent Picklist Considerations

User Permissions Needed	
To define and edit dependent picklists:	“Customize Application”

Consider the following when defining dependent picklists:

Checkboxes

Checkbox fields can be controlling fields but not dependent fields.

Converting fields

Convert your existing fields to dependent picklists or controlling fields without affecting the existing values in your records. Going forward, dependency rules apply to any changes to existing records or new records.

Default values

You can set default values for controlling fields but not for dependent picklists.

Field-level security

Field-level security settings for a controlling field and dependent picklist are completely independent. Remember to hide controlling fields whenever its correlating dependent picklist is hidden.

Multi-select picklists

Multi-select picklists can be dependent picklists but not controlling fields.

Editing Dependent Picklists

User Permissions Needed	
To edit field dependencies:	“Customize Application”

To edit dependent picklists:

1. Navigate to the appropriate object:

- Click **Create > Objects**, and click the name of the custom object in the list.
2. Click **Field Dependencies**.
 3. Click **Edit** next to the field dependency relationship you want to change.
 4. Use the [field dependency matrix](#) to specify the dependent picklist values that are available when a user selects each controlling field value.
 5. Optionally, click **Preview** to test your selections.
 6. Click **Save**.

Deleting Picklist Dependencies

User Permissions Needed	
To delete picklist dependencies:	“Customize Application”

If you no longer want the values of a dependent picklist to depend on a controlling field, delete its dependency. Deleting the dependency removes the logic that defines how the values of the picklist depend on the controlling field, but doesn't delete the fields or affect their data.

To delete picklist dependencies:

1. Click **Create > Objects**, and click the name of the custom object in the list.
2. Click **Field Dependencies**.
3. Click **Del** next to the field dependency relationship you want to delete.
4. Click **OK** to confirm.

Managing Roll-Up Summary Fields

About Roll-Up Summary Fields

While your formula fields calculate values using fields within a single record, roll-up summary fields calculate values from a set of related records, such as those in a related list. You can create roll-up summary fields that automatically display a value on a master record based on the values of records in a detail record. These detail records must be directly related to the master through a master-detail relationship.

You can perform different types of calculations with your roll-up summary fields. You can count the number of detail records related to a master record, or calculate the sum, minimum value, or maximum value of a field in the detail records.

Before you begin creating roll-up summary fields for your organization, review the implementation tips and best practices.

Implementation Tips

Administration

- Create roll-up summary fields on:
 - ◊ Any custom object that is on the master side of a master-detail relationship
- The types of fields you can calculate in a roll-up summary field depend on the type of calculation. For example,
 - ◊ Number, currency, and percent fields are available when you select SUM as the roll-up type.
 - ◊ Number, currency, percent, date, and date/time fields are available when you select MIN or MAX as the roll-up type.
- You may not be able to change the field type of a field that you reference in a roll-up summary field.

- Make sure that the filter for your roll-up summary does not encounter a formula field that results in “#Error!”. If your filter criteria uses a formula field that results in an error, no matches are returned for that filter criterion. For example, if your roll-up summary filter is “Formula Field equals 10” and two records contain errors while one contains the value “10” in that field, your summary only includes the record with the value “10.”
- Long text area, multi-select picklist, Description, system fields like Last Activity, cross-object formula fields, and lookup fields (such as the Product Code field) cannot be used in the field column of roll-up summary filters.
- Auto number fields are not available as the field to aggregate in a roll-up summary field.
- After you have created a roll-up summary field on an object, you cannot convert the object's master-detail relationship into a lookup relationship.

Management

- Roll-up summary fields can calculate the values of formula fields if they do not contain cross-object field references or functions that automatically derive values on the fly, such as **NOW** or **TODAY**.



Note: The value of a formula field can result in “#Error!”, which affects the summarized total. If your roll-up summary type is COUNT, records are included regardless of whether they contain a formula field with an error, but when the Field to Aggregate is a formula field that results in “#Error!”, calculations of type MIN, MAX, and SUM exclude those formula values.

- Roll-up summary fields can trigger workflow rules and field validations. However, workflow rules and field validations do not fire when the following changes cause a mass recalculation of roll-up summary values:
 - ◊ Changing the roll-up summary definition (such as the object, function, or field being aggregated)
 - ◊ Changing the expression of a formula field referenced in a roll-up summary field
 - ◊ Replacing picklist values for picklist fields referenced in the roll-up summary filter
 - ◊ Changing picklist record type definitions
 - ◊ Changing currency conversion rates
- Calculating roll-up summary field values may take up to 30 minutes, depending on the number of records affected and other factors.
- You are not prevented from creating roll-up summary fields that might result in invalid values, such as February 29th in a non-leap year. If an existing roll-up summary field results in an invalid value, the value is not recalculated, and the field will continue to display with an invalid roll-up summary icon (☒) until you change the values being summarized.
- If your organization uses multiple currencies, the currency of the master record determines the currency of the roll-up summary field. For example, if the master and detail records are in different currencies, the values in the detail record are converted into the currency of the master record before calculating the final value and displaying it in the same currency as the master record.
- Database.com will prevent users from saving a record when doing so would invalidate a related record. For example, if the related master record has a validation rule that requires the roll-up summary field value to be greater than 100, and the user's change to the child record will put the value over 100, the user will not be able to save the record.

Best Practices

- Apply field-level security to your roll-up summary fields if they calculate values that you do not want visible to users. Fields that your users cannot see due to field-level security settings on the detail record are still calculated in a roll-up summary field.
- If you have **validation rules**, consider how they affect roll-up summary fields. A validation error can display when saving either the detail or master record because the value in a roll-up summary field changes when the values in the detail records change.
- Because roll-up summary fields are not displayed on edit pages, you can use them in **validation rules** but not as the error location for your validation.

- Avoid referencing a roll-up summary field from a child record. The roll-up summary field will have the previous value because the parent record has not been updated. If you reference a roll-up summary field from a parent record, the roll-up summary field will always have the new value because that rule runs after the parent value has been updated.

If you are trying to enforce a record limit of 25 on the parent roll-up summary field, when you add a new child record, your validation rule on the child object needs to check if the count is already 25 or greater.

```
AND (ISNEW(), Sample.Line_Count__c >= 25)
```

- Plan your implementation of roll-up summary fields carefully before creating them. Once created, you cannot change the detail object selected or delete any field referenced in your roll-up summary definition.
- Automatically derived fields such as current date or current user are not allowed in roll-up summary fields. This includes formula fields containing functions that automatically derive values on the fly, such as **NOW** and **TODAY**. Formula fields that include related object merge fields are also not allowed in roll-up summary fields.

Defining Roll-Up Summaries

User Permissions Needed	
To view roll-up summary field definitions:	“View Setup and Configuration”
To edit roll-up summary field definitions:	“Customize Application”

Define roll-up summary fields on the object that is on the master side of a master-detail relationship. If a relationship does not already exist, first create a [master-detail relationship](#) between the master object that displays the value and the detail object containing the records you are summarizing.

To define a roll-up summary field:

- Create a custom field on the object where you want the field displayed. Summary fields summarize the values from records on a related object, so the object on which you create the field should be on the master side of a master-detail relationship. For instructions on creating a custom field, see [Creating Custom Fields](#) on page 98.
- Choose the **Roll-Up Summary** field type, and click **Next**.
- Enter a field label and any other attributes. Click **Next**. For information on the attributes you can set, see [Custom Field Attributes](#) on page 101.
- Select the object on the detail side of a master-detail relationship. This object contains the records you want to summarize.
- Select the type of summary:

Type	Description
COUNT	Totals the number of related records.
SUM	Totals the values in the field you select in the Field to Aggregate option. Only number, currency, and percent fields are available.
MIN	Displays the lowest value of the field you select in the Field to Aggregate option for all directly-related records. Only number, currency, percent, date, and date/time fields are available.
MAX	Displays the highest value of the field you select in the Field to Aggregate option for all directly-related records. Only number, currency, percent, date, and date/time fields are available.

6. Enter your filter criteria if you want a selected group of records in your summary calculation. If your organization uses multiple languages, enter filter values in your organization's default language. For more information on using filter criteria, see [Entering Filter Criteria](#) on page 81.
7. Click **Next**.
8. Set the field-level security to determine whether the field should be visible for specific profiles, and click **Next**.
9. Click **Save** to finish or **Save & New** to create more custom fields.

Defining Universally Required Fields

About Universally Required Fields

A universally required field is a custom field that must have a value whenever a record is saved. A universally required field is a custom field that must have a value whenever a record is saved. Making a field required through field-level security ensures users must enter a value, but making a field required universally gives you a higher level of data quality.

You can make the following types of custom fields universally required:

- Currency
- Date
- Date/Time
- Email
- Master-Detail Relationship (always required)
- Number
- Percent
- Phone
- Text
- Text Area
- URL

To make a custom field universally required, select the **Required** checkbox when defining the custom field.

Considerations for Universally Required Fields

Review the following considerations before making your custom fields universally required.

- For a list of the types of custom fields that you can make universally required, see [About Universally Required Fields](#) on page 193.
- Edit pages always display universally required fields, regardless of field-level security.
- [Default values](#) should not be assigned to fields that are both [required](#) and [unique](#), as uniqueness errors may result.
- You cannot make a field universally required if it is used by a field update that sets the field to a blank value. For details, see [Defining Field Updates](#) on page 416.
- Required fields may be blank on records that existed before making the field required. When a user updates a record with a blank required field, the user must enter a value in the required field before saving the record.

Using Rich Text Area Fields

User Permissions Needed	
To create or change custom fields:	"Customize Application"

To improve the appearance of text, including adding images and hyperlinks, create rich text area custom fields.

Implementation Tips

Before creating rich text area custom fields, note the following:

- Database.com supports up to 32,768 characters for each rich text area field, including the HTML tags. If desired, you can set a lower limit.
- [Deleting a rich text area field](#) moves it to the Deleted Fields section on the custom object.
- The field limit for rich text area fields is 25. Additionally, each object can contain a total of 1.6 million characters across long text area and rich text area fields. The default character limit for long text area and rich text area fields is 32,000 characters. A long text area or rich text area field needs to contain at least 256 characters.
- Maximum size for images uploaded through the API is 1 MB. Only .gif, .jpg and .png file types are supported.
- You can only convert rich text area fields into long text area fields. Any images are deleted the next time the long text area field is saved.
- You can't add a hyperlink to an image.
- There is no support for disabling specific rich text area features. For example, you can't disable support for hyperlinks or images on certain fields.
- JavaScript or CSS is treated as text.
- When a rich text area field is used in a formula, the HTML tags are stripped out before the formula is run. For example, when a rich text area field is used in a validation rule's criteria, the HTML tags are removed before the evaluation.
- The text part of rich text area fields counts towards [data storage](#) for the entity that contains the field. The uploaded images within the rich text area fields are counted towards [file storage](#) for the entity that contains the field.
- The rich text area field is available in the API.

Best Practices

- You can specify how big the editor box should be for this field by configuring the “Number of lines displayed” property in the field’s setup.

Managing Users

User Management Overview

In Database.com, every user is identified by a username, password, and a single profile. The profile determines what tasks users can perform, what data they see, and what they can do with the data.

By default, the Database.com Edition provides three Database.com Admin Licenses and three Database.com User licenses. Database.com Admin license users can make changes to Database.com schemas or other metadata using the point-and-click tools in the Database.com Console. Users with Database.com User licenses have API access to data stored in Database.com. Additionally, you can obtain Database.com Light User licenses from salesforce.com. Users with Database.com Light User licenses have API access only, can belong to Chatter groups (but no other groups) and not roles or queues.

As an administrator, you can perform user management tasks such as creating and editing users, and resetting passwords. You can also grant permissions, create and manage other types of users, and create custom fields.

You can control a user's access to data in several ways:

- To control access to applications and objects, including fields and record types within objects, use profiles and permission sets.
- To control access to specific records, use sharing settings and rules.

Managing User Tasks

Viewing and Managing Users

User Permissions Needed	
To view user lists :	“View Setup and Configuration”

To view and manage the users in your organization, click **Manage Users > Users**. The user list shows all the users in your organization. From the user list, you can:

- Show a filtered list of users by selecting a list from the View drop-down list.
- [Edit or create custom list views](#). For example, create a view with search criteria of “Profile Name equals Read Only” to show all users with the Read Only profile.
- [Create one or multiple users](#).
- [Reset passwords for selected users](#).
- [Edit a user](#).
- View a user's detail page by clicking the name, alias, or username.
- View or edit a [profile](#) by clicking the profile name.



Tip: To create custom fields for users, click **Customize > Users > Fields**

Tips for Managing Users

- You can create custom fields for users to display on the user detail page. To access these options, click **Customize > Users**.
- You can use the sidebar search to search for any user in your organization, regardless of the user's status. However, when using a lookup dialog from fields within records, the search results return only active users.
- To simplify user management in organizations with large numbers of users, [delegate aspects of user administration to non-administrator users](#).

Adding New Users

User Permissions Needed	
To create new users:	“Manage Users”

By default, the Database.com Edition provides three Database.com Admin licenses and three Database.com User licenses.

To create a new user for your organization:

1. Click **Manage Users > Users**.
2. Click **New User**.
3. Enter the user's first name, last name, and email address. The email address becomes the username.



Note: If the user's name includes non-English characters, the user must add the specified language to the mail format settings within Outlook if viewing email in Outlook.

4. Select a [User License](#). The profiles available to you depend on the user license you choose.
 5. Select a profile. For more information, see [User Profiles Overview](#) on page 213.
 6. Select a role from the list defined for the **Role** field.
- See [Overview of Roles](#) on page 203 if you have not set up the role hierarchy.
7. Check **Generate new password and notify user immediately** to have the user's login name and a temporary password emailed to the new user.

Editing Users

User Permissions Needed	
To edit users:	“Manage Users”

To view or edit user information:

1. Click **Manage Users > Users**.

You can also edit your organization's users from the Users in Role related list in the role detail page.

2. Click **Edit** next to a user's name.
3. Change the necessary information and click **Save**.

See [Adding New Users](#) for more detail about the fields on the user information page.

Tips for Editing Users

- Users can change or add to their own [personal information](#) after they log in.
- If you change a user's email address and **Generate new password and notify user immediately** is deselected, a confirmation message will be sent to the new email address that you entered. The user must click the link provided in that message for the new email address to take effect. This process ensures system security. When generating a new password for a user, the new password is automatically sent to the user's email address and email verification is not enforced.
- If you change a user's username, a confirmation email with a login link is sent to the email address associated with that user account. If the user has problems logging into future sessions, they can use the link in the email. Problems might occur because an organization could have multiple login servers. It can sometimes take up to 24 hours for the username change to replicate to all of them. The link in the email connects directly to the server where the actual username change was made. This ensures that the user can always login even if server replication is slow.
- To [deactivate users](#) so they can no longer access your organization, deselect the **Active** box.
- Administrators can restrict the domain names of users' email addresses to a list of explicitly allowed domains. Any attempts to set an email address with another domain will result in an error message. Contact salesforce.com to enable this functionality for your organization.
- Click **Unlock** to unlock a user that is locked out of Database.com. This button is only available when a user is locked out.

- Click **Grant Checkout Access** to give a user access to **Checkout**. Using Checkout, the user can purchase Database.com licenses, and other related products. Additionally, within Checkout, the user can view the organization's quotes, installed products, orders, invoices, payments, and contracts.

Understanding User License Types

User Permissions Needed
To view user license types: "View Setup and Configuration"

You may have more than one type of user license in your organization. A user license entitles a user to different functionality within Database.com and determines which profiles and permission sets are available to the user.

To view a list of the active user licenses in your company, click **Company Profile > Company Information**. This page lists the following for each type of user license:

- Status indicates the status of the license.
- Total Licenses indicates the number of licenses for which your company is billed and that are available to you.
- Used Licenses is the number of licenses that you have assigned to users.
- Remaining Licenses is the number of unused licenses.

If Checkout is enabled for your organization, you can click **Buy More Licenses** to go to Checkout to buy additional user licenses.



Note: You may see other types of licenses listed on this page if your organization has purchased custom user licenses for different types of functionality. Your organization may also have other licenses that are still supported, but are no longer available. Contact salesforce.com for more information.

Database.com License Types

User License	Description	Default Number of Available Licenses
Database.com Admin	Designed for users who need to administer Database.com, or make changes to Database.com schemas or other metadata using the point-and-click tools in the Database.com Console.	Database.com Edition: 3
Database.com User	Designed for users who need API access to data stored in Database.com.	Database.com Edition: 3 Contact salesforce.com to obtain Database.com User Licenses
Database.com Light User	Designed for users who need only API access to data, need to belong to Chatter groups (but no other groups), and don't need to belong to roles or queues. Access to data is determined by organization-wide sharing defaults.	Database.com Edition: 0 Contact salesforce.com to obtain

User License	Description	Default Number of Available Licenses
		Database.com Light User Licenses

Deactivating Users

User Permissions Needed	
To deactivate users:	“Manage Users”

You can't completely delete users from the system, but you can deactivate their logins so that they can no longer use the service.

1. Click **Manage Users > Users**.
2. Click **Edit** next to a user's name.
3. Deselect the **Active** checkbox and click **Save**.

Tips on Deactivating Users

Consider the following when deactivating users:

- Deactivated users lose access to any records that were manually shared with them, or records that were shared with them as team members. However, you can still transfer their data to other users and view their names on the Users page.
- A deactivated user doesn't count against your organization's available user licenses. However, deactivating a user doesn't reduce the number of licenses for which your organization is billed; you must change your organization's license count to change your billing.
- You can't deactivate a user selected in a custom hierarchy field even if you delete the field. You must delete and permanently erase the field first. For more information, see [Managing Deleted Custom Fields](#) on page 106.
- It's possible for inactive users to be listed as “Created By” users even though they are no longer active in an organization. This can happen because some system operations create records and toggle preferences, acting as an arbitrary administrator user in your organization to complete the task. This user may be active or inactive.

Managing Passwords

About Passwords

User Permissions Needed	
To set password policies:	“Manage Users”
To reset user passwords and unlock users:	“Reset User Passwords and Unlock Users”

There are several settings you can configure to ensure that your users' passwords are strong and secure:

- Password policies—set various password and login policies, such as specifying an amount of time before all users' passwords expire, the level of complexity required for passwords, and so on. See [Setting Password Policies](#) on page 199.
- User password expiration—expire the passwords for all the users in your organization, except for users with “Password Never Expires” permission. See [Expiring Passwords](#) on page 202.

- User password resets—reset the password for specified users. See [Resetting Passwords](#) on page 201.
- Login attempts and lockout periods—if a user is locked out of Database.com due to too many failed login attempts, you can unlock them. See [Editing Users](#) on page 196.

Password Requirements

A password cannot contain your User Name and cannot match your first or last name.

A new organization has the following default password requirements:

- A password must contain at least eight characters.
- A password must contain at least one alphabetic character and one number.
- The answer to the question posed if you forget your password cannot contain your password.
- The last three passwords are remembered and cannot be reused when you are changing your password.

The password policies, including these defaults, can be updated. See [Setting Password Policies](#) on page 199 for more details.

Setting Password Policies

User Permissions Needed	
To set password policies:	“Manage Users”

For your organization’s security, you can set various password and login policies.



Note: User passwords cannot exceed 16,000 bytes.

1. Click **Security Controls > Password Policies**.
2. Customize the password settings.

Field	Description
User passwords expire in	The length of time until all user passwords expire and must be changed. Users with the “Password Never Expires” permission are not affected by this setting. The default is 90 days.
Enforce password history	Save users’ previous passwords so that they must always reset their password to a new, unique password. Password history is not saved until you set this value. The default is 3 passwords remembered. You cannot select No passwords remembered unless you select Never expires for the User passwords expire in field.
Minimum password length	The minimum number of characters required for a password. When you set this value, existing users aren’t affected until the next time they change their passwords. The default is 8 characters.

Field	Description
Password complexity requirement	<p>The restriction on which types of characters must be used in a user's password.</p> <p>Complexity levels:</p> <ul style="list-style-type: none"> • No restriction—allows any password value and is the least secure option. • Must mix alpha and numeric—requires at least one alphabetic character and one number. This is the default. • Must mix alpha, numeric, and special characters—requires at least one alphabetic character, one number, and one of the following characters ! # \$ % - _ = + < >.
Password question requirement	<p>The values are Cannot contain password, meaning that the answer to the password hint question cannot contain the password itself; or None, the default, for no restrictions on the answer. The user's answer to the password hint question is required.</p>
Maximum invalid login attempts	<p>The number of login failures allowed for a user before they become locked out.</p>
Lockout effective period	<p>The duration of the login lockout. The default is 15 minutes.</p> <p> Note: If users are locked out, they must wait until the lockout period expires. Alternatively, a user with the "Reset User Passwords and Unlock Users" permission can unlock them by clicking Manage Users > Users, selecting the user, then clicking Unlock. This button is only available when a user is locked out.</p>

3. Customize the forgotten password and locked account assistance information.

Field	Description
Message	<p>When set, this custom message appears in the Account Lockout email and at the bottom of the Confirm Identity screen for users resetting their passwords. You can customize it with the name of your internal help desk or a system administrator. For the lockout email, the message only appears for accounts that need an administrator to reset them. Lockouts due to time restrictions get a different system email message.</p>

Field	Description
Help link	<p>If set, this link displays with the text defined in the Message field. In the Account Lockout email, the URL displays just as it is typed into the Help link field, so the user can see where the link takes them.</p> <p>On the Confirm Identity password screen, the Help link URL combines with the text in the Message field to make a clickable link.</p> <p>Valid protocols:</p> <ul style="list-style-type: none"> • http • https • mailto:

4. Specify an alternative home page for users with the “API Only User” permission. After completing user management tasks such as resetting a password, API-only users are redirected to the URL specified here, rather than to the login page.
5. Click **Save**.

Resetting Passwords

User Permissions Needed	
To reset passwords:	“Reset Passwords and Unlock Users”

To reset a user’s password:

1. Click **Manage Users > Users**.
2. Select the checkbox next to the user’s name. Optionally, to change the passwords for all currently displayed users, check the box in the column header to select all rows.
3. Click **Reset Password** to have a new password emailed to the user.

Tips on Resetting Passwords

Consider the following when resetting passwords:

- Only an administrator can reset Single Sign-On user passwords; Single Sign-On users cannot reset their own passwords. For information about Single Sign-On, see [About Single Sign-On](#) on page 289.
- Resetting users’ passwords, might cause them to activate their computers to successfully log in to Database.com. For more information, see [Activating Your Computer](#) on page 78.
- Resetting locked-out users’ passwords automatically unlocks their accounts as well.
- When users lose their passwords, they can click the **Forgot your password?** link on a failed login page to receive a new password via email. They must correctly answer a previously defined security question before they can reset their password and log in. You can customize part of the page where the user answers the security question with additional information about where to go to for help. See [Setting Password Policies](#) on page 199.



Note: If the user has not defined a security question, or fails to answer correctly when trying to login, the password is not reset.

Expiring Passwords

User Permissions Needed	
To expire all passwords:	“Manage Users”

To expire passwords for all users, except those with the “Password Never Expires” permission:

1. Click **Security Controls > Expire All Passwords**.
2. Select the **Expire all user passwords** checkbox.
3. Click **Save**.

The next time each user logs in, he or she will be prompted to reset his or her password.

Tips on Expiring Passwords

Consider the following when expiring passwords:

- After you expire passwords, users might need to activate their computers to successfully log in to Database.com. See [Activating Your Computer](#) on page 78.
- You can expire passwords for all users any time you want to enforce extra security for your organization. For more options you can set to ensure password security, see [Setting Password Policies](#) on page 199.

Logging In as Another User

User Permissions Needed	
To log in as another user who has granted you access:	“Manage Users” AND “Modify All Data”

To assist other users, administrators can log in to Database.com as another user if that user has [granted login access](#).

To log in as another user:

1. Click **Manage Users > Users**.
2. Click the **Login** link next to the user who has granted you access. The **Login** link and button only appear for users who have granted login access to an administrator.
3. Click **Your Name > Logout** to return to your administrator account.

Note: You can also log in as another user from the [user detail](#) page and the [Users in Role](#) list page.



Granting Checkout Access

User Permissions Needed	
To grant Checkout access:	“Manage Billing”
To edit users:	“Manage Users”

Users with the “Manage Billing” permission automatically have access to Checkout when it is enabled for your organization. These users can also grant access to other users within your organization.

To give a user access to Checkout:

1. Click **Manage Users > Users**.
2. Click on the appropriate user's name to open the user detail page.
3. Click **Edit**.
4. Select the **Checkout Enabled** checkbox. The user is notified by email when his or her Checkout account is activated and available for login.

Controlling Login Access

User Permissions Needed	
To control login access policies:	“Manage Users”

By default, users can always grant login access to Database.com. You may choose to enable only administrators to grant login access to Database.com.

1. Choose **Security Controls > Login Access Policies**.
2. Click **Available to Administrators Only** next to Salesforce.com Support.
3. Click **Save**.

Managing Roles

Overview of Roles

User Permissions Needed	
To create, edit, and delete roles:	“Manage Users”
To assign users to roles:	“Manage Users”

Depending on your sharing settings, roles can control the level of visibility that users have into your organization's data. Users at any given role level can view and edit all data owned by or shared with users below them in the hierarchy, unless your organization's sharing model for an object specifies otherwise. Specifically, in the [Organization-Wide Defaults related list](#), if the **Grant Access Using Hierarchies** option is disabled for a custom object, only the record owner and users granted access by the organization-wide defaults receive access to the object's records.

Working with Roles

To view and manage your organization's roles, click **Manage Users > Roles**.

- Choose one of the following list view options:

Show in tree view

See a visual representation of the parent-child relationships between your roles. Click **Expand All** to see all roles, or **Collapse All** to see only top-level roles. To expand or collapse an individual node, click the plus (+) or minus (-) icon.

Show in sorted list view

See a list that you can sort alphabetically by role name, parent role (**Reports to**), or report display name. If your organization has a large number of roles, use this view for easy navigation and filtering.

To show a filtered list of items, select a predefined list from the View drop-down list, or click **Create New View** to define your own [custom view](#). To edit or delete any view you created, select it from the View drop-down list and click **Edit**.

Show in list view

See a list of roles and their children, grouped alphabetically by the name of the top-level role. The columns are not sortable. This view is not available for hierarchies with more than 1,000 roles.

- To create a role, click **New Role** or **Add Role**, depending whether you are viewing the list view or tree view of roles, then edit the role fields as needed. You can create up to 500 roles for your organization.
- To edit a role, click **Edit** next to a role name, then update the [role fields](#) as needed.
- To delete a role, click **Delete** next to the role name.
- To [assign other users to a role](#), click **Assign** next to the role name.
- To view detailed information about a role, click a role name.



Tip: To simplify user management in organizations with large numbers of users, enable [delegated administrators](#) to manage users in specified roles and all subordinate roles.

Notes on Roles

- Every user must be assigned to a role since you will not be able to see data created by users not assigned to roles.
- All users that require visibility to the entire organization should belong to the highest level in the hierarchy.
- It is not necessary to create individual roles for each title at your company, rather you want to define a hierarchy of roles to control access of information entered by users in lower level roles.
- When you change a user's role, any relevant sharing rules are evaluated to add or remove access as necessary.
- Users that gain access to data due to their position in hierarchies do so based on a setting in your [organization-wide defaults](#).

Viewing and Editing Roles

User Permissions Needed	
To view role details:	“View Setup and Configuration”
To edit and delete roles:	“Manage Users”
To view users:	“View Setup and Configuration”
To edit users:	“Manage Users”

To view detailed information about a role, click **Manage Users > Roles**, and click the role name.

In the Role Detail related list:

- To view the role detail page for a parent or sibling role, click the role name in the Hierarchy or Siblings list.
- To edit the role details, click **Edit**.
- To remove the role from the hierarchy, click **Delete**.

In the Users in Role related list:

- To assign a user to the role, click **Assign Users to Role**.
- To add a user to your organization, click **New User**.
- To modify user information, click **Edit** next to a user name.
- To view a user's details, click the user's full name, alias, or username.

When Active is selected, the user can log into Database.com. **Deactivated users**, such as employees who are no longer with your company, cannot log in to Database.com.



Note: When you edit groups and roles, sharing rules are automatically reevaluated to add or remove access as needed. If these changes affect too many records at once, a warning appears that the sharing rules won't be automatically reevaluated, and you must manually recalculate them.

Assigning Users to Roles

User Permissions Needed	
To assign users to roles:	“Manage Users”

To quickly assign users to a particular role:

1. Click **Manage Users > Roles**.
2. Click **Assign** next to the name of the desired role.



Note: You can also access this page by clicking **Assign Users to Role** from the Users in Role related list.

3. Make a selection from the drop-down list to show the available users.
4. Select a user on the left, and click **Add** to assign the user to this role.



Note: Removing a user from the Selected Users list deletes the role assignment for that user.

Viewing Role Sharing Groups

User Permissions Needed	
To view users:	“View Setup and Configuration”
To edit users:	“Manage Users”

For each [role](#) in your hierarchy Database.com automatically creates sharing groups, which you can use in sharing rules and manual sharing:

- Role—users in the role plus users in roles above it in the hierarchy
- Role and Subordinates—users in the role plus users in roles above and below it in the hierarchy

To view sharing group members:

1. For roles, click **Manage Users > Roles**.
2. Click the role name.
3. Click a link in the **Sharing Groups** field.

From this page, you can view detailed user information, edit user information, and access related information.

- To show a filtered list of items, select a predefined list from the View drop-down list, or click **Create New View** to define your own [custom view](#). To edit or delete any view you created, select it from the View drop-down list and click **Edit**.
- Click **Edit** next to a username to [edit the user information](#).
- Click **Login** next to a username to [log in as that user](#). This link is only available if the user has granted you login access.

Viewing Users in Role Lists

User Permissions Needed	
To view users:	“View Setup and Configuration”
To edit users:	“Manage Users”

The Users in Role list page displays a list of users assigned to the selected role. From this page, you can view detailed user information, edit users, and access other related information.

- To show a filtered list of items, select a predefined list from the View drop-down list, or click **Create New View** to define your own [custom view](#). To edit or delete any view you created, select it from the View drop-down list and click **Edit**.
- Click **Edit** next to a user name to [edit the user information](#).
- Click **Login** next to a user name to [log in as that user](#). This link is only available for users who have granted login access to an administrator.

Managers in the Role Hierarchy

The Managers in the Role Hierarchy related list shows all of the users above you in the hierarchy. These users have the same access to your data as you do—they have access to all data you own or that has been shared with you.

To view this related list, click **My Personal Information > Personal Information**, and scroll down to the related list.

Role Fields

User Permissions Needed	
To create or edit roles:	“Manage Users”

The following fields (listed in alphabetical order) make up a role entry. Some of these fields may not be visible or editable depending on your organization's permissions and sharing settings.

Field	Description
Label	The name used to refer to the role or title of position in any user interface pages, for example, Western Sales VP.
Modified By	The name of the user who last modified this role's details, and the date and time that the role was modified.
Role Name	The unique name used by the API. The name must begin with a letter and use only alphanumeric characters and underscores. The name cannot end with an underscore or have two consecutive underscores.
Role Name as displayed on reports	A role name that appears in reports. When editing a role, if the Role Name is long, you can enter an abbreviated name in this field.
Sharing Groups	These groups are automatically created and maintained. The Role group contains all users in this role plus all users in roles above this role. The Role and Subordinates group contains all users in this role plus all users in roles above and below this role in the hierarchy.
This role reports to	The role above this role in the hierarchy.

Managing User Permissions and Access

Overview of User Permissions and Access

User permissions and access settings specify what users can do within an organization. For example, permissions determine a user's ability to edit an object record or reset a user's password. Access settings determine other functions, such as access to Apex classes and the hours when users can log in.

Permissions and access settings are specified in user profiles and permission sets. Every user is assigned only one profile, but can also have multiple permission sets.

When determining access for your users, it's a good idea to use profiles to assign the minimum permissions and access settings for specific groups of users, then use permission sets to grant additional permissions.

Because you can assign many permission sets to users and permission sets are reusable, you can distribute access among more logical groupings of users, regardless of their primary job function. For example, you could create a permission set that gives read access to a custom object and assign it to a large group of users, and create another permission set that gives edit access to the object and assign it to only a few users. You can assign these permission sets to various types of users, regardless of their profiles.

The following table shows the types of permissions and access settings that are specified in profiles and permission sets. Some profile settings aren't included in permission sets.

Permission or Setting Type	In Profiles?	In Permission Sets?
Object permissions	✓	✓
Field permissions	✓	✓
User permissions (app and system)	✓	✓
Apex class access	✓	✓
Service provider access (if Database.com is enabled as an identity provider)	✓	✓
Login hours	✓	
Login IP ranges	✓	

See Also:

- [Permission Sets Overview](#)
- [User Profiles Overview](#)
- [Revoking Permissions and Access](#)

Managing Permission Sets

Permission Sets Overview

A permission set is a collection of settings and permissions that give users access to various tools and functions. The settings and permissions in permission sets are also found in profiles, but permission sets extend users' functional access without changing their profiles. For example, to give users access to a custom object, create a permission set, enable the required permissions for the object, and assign the permission set to the users. You never have to change profiles, or create a profile for a single use case. While users can have only one profile, they can have multiple permission sets.

Permission sets include:

- Object and field permissions
- App permissions (not available in some permission sets)
- Apex class access

- System permissions
- Service providers (only if you've enabled Database.com as an identity provider)

Creating Permission Sets

User Permissions Needed	
To create permission sets:	“Manage Users”

You can either clone an existing permission set or create a new one. A cloned permission set starts with the same user license and enabled permissions as the permission set it is cloned from, while a new permission set starts with no user license selected and no permissions enabled. You can create up to 1,000 permission sets.

1. Click **Manage Users > Permission Sets**.

2. Do one of the following:

- To create a permission set with no permissions enabled, click **New**.
- To create a permission set based on an existing set, click **Clone** next to the set you want to copy. You can also select the permission set and click **Clone** in the overview or one of the settings pages.



Note: Clone a permission set only if the new one will have the same user license as the original. In a cloned permission set, you can't select a different license. Clone only permission sets with a Database.com license. Permission sets with other licenses can't be assigned to users.

3. Enter a label, API name, and description.

The API name is a unique name used by the API. It must begin with a letter, and use only alphanumeric characters and underscores. It can't include spaces, end with an underscore, or have two consecutive underscores.

4. If this is a new permission set, select the user license that matches the users who will use this permission set. For example, if you plan to assign this permission set to users with the Database.com User license, select Database.com User. Select only a Database.com license. Permission sets with other licenses can't be assigned to users.

5. Click **Save**.

The permission set overview page appears. From here you can navigate to the permissions you want to add or change.

Creating and Editing Permission Set List Views

User Permissions Needed	
To create, edit, and delete permission set list views:	“Manage Users”

You can create and edit permission set list views to show a list of permission sets with specific fields and permissions. For example, you could create a list view of all permission sets in which “Modify All Data” is enabled.

1. In the Permission Sets page, click **Create New View**, or select a view and click **Edit**.
2. Enter the view name.
3. Under Specify Filter Criteria, specify the conditions that the list items must match, such as `Modify All Data equals True`.
 - a. Type a setting name, or click the lookup icon to search for and select the setting you want.

- b. Choose a filter operator.
 - c. Enter the value that you want to match.
 - d. To specify another filter condition, click **Add Row**. You can specify up to 25 filter condition rows.
To remove a filter condition row and clear its values, click .
4. Under Select Columns to Display, specify the settings that you want to appear as columns in the list view. You can add up to 15 columns.
- a. From the Search drop-down list, select a setting type.
 - b. Enter part or all of a word in the setting you want to add and click **Find**.
-  **Note:** If the search finds more than 500 values, no results appear. Refine your search criteria to show fewer results.
- c. To add or remove columns, select one or more column names and click the **Add** or **Remove** arrow.
 - d. Use the **Top**, **Up**, **Down**, and **Bottom** arrows to arrange the columns in the sequence you want.
5. Click **Save**, or if you're cloning an existing view, rename it and click **Save As**.

Editing Permission Sets from a List View

User Permissions Needed	
To edit multiple permission sets from the list view:	“Manage Users”
	AND
	“Customize Application”
	AND
	“Mass Edits from Lists”

You can change permissions in up to 200 permission sets directly from the list view, without accessing individual permission sets.



Note: Use care when editing permission sets with this method. Making mass changes may have a widespread effect on users in your organization.

1. Select or create a list view that includes the permission sets and permissions you want to edit. Editable fields display a pencil icon () when you hover over the field, while non-editable fields display a lock icon ()
2. To edit multiple permission sets, select the checkbox next to each one you want to edit. If you select permission sets on multiple pages, the selections on each page are remembered.
3. Double-click the permission you want to edit. For multiple permission sets, double-click the permission in any of the selected permission sets.
4. In the dialog box that appears, enable or disable the permission. In some cases, changing a permission may also change other permissions.
5. To change multiple permission sets, select **All *n* selected records** (where *n* is the number of permission sets you selected).
6. Click **Save**.

If you edit multiple permission sets, only those that support the permission you are changing will change. For example, if you use inline editing to enable “Modify All Data” in a permission set, but because of its user license the permission set doesn’t have “Modify All Data,” the permission set won’t change.

If any errors occur, the error message lists each permission set and a description of the error. Click the permission set name to open its overview page. The permission sets you’ve clicked appear in the error window in gray, strike-through text.



Note: To view the error console, pop-up blockers must be disabled for the Database.com domain. To check if your browser allows pop-up windows, click **My Personal Information > Reminders**, and then click **Preview Reminder Alert**.

Any changes you make are recorded in the setup audit trail.

Working in a Permission Set's Overview Page

User Permissions Needed	
To view permission sets:	“View Setup and Configuration”
To delete permission sets and edit permission set properties:	“Manage Users”

A permission set’s overview page provides an entry point for all of the permissions in a permission set. To open a permission set overview page, click **Manage Users > Permission Sets** and select the permission set you want to view.

From the permission set overview page, you can:

- [Search for an object, setting, or permission](#)
- [Create a permission set based on the current permission set](#)
- If it’s not assigned to any users, remove the permission set by clicking **Delete**
- Change the permission set label, API name, or description by clicking **Edit Properties**
- View a list of users who are assigned to the permission set
- View or edit:
 - ◊ Object and field permissions
 - ◊ App permissions (not available in some permission sets)
 - ◊ Apex class access settings
 - ◊ System permissions
 - ◊ Service providers (if you’ve enabled Database.com as an identity provider)

About App and System Settings in Permission Sets

In permission sets, permissions and settings are organized into app and system categories, which reflect the rights users need to administer and use system and app resources.

App Settings

The Apps section of the permission sets overview page contains settings that are directly associated with the business processes that the apps enable. The Apps section contains links to:

- Object and field permissions
- App permissions (not available in some permission sets)
- Apex class access

System Settings

Some system functions apply to an organization and not to any single app. For example, “View Setup and Configuration” allows users to view setup and administrative settings pages. Other system functions apply to all apps. In some cases, such as with “Modify All Data,” a permission applies to all apps, but also includes non-app functions, like the ability to download the Data Loader.

In the permission sets overview page, the System section contains links to:

- System permissions
- Service providers (if you've enabled Database.com as an identity provider)

Searching Permission Sets

User Permissions Needed	
To search permission sets:	“View Setup and Configuration”

On any of the detail pages, type at least three consecutive letters of an object, setting, or permission name in the **Find Settings...** box. The search terms aren't case-sensitive. As you type, suggestions for results that match your search terms appear in a list. Click an item in the list to go to its settings page. You can search for:

Item	Example
Objects	Type an existing object's name. For example, let's say you have an Albums custom object, type <code>albu</code> , then select <code>Albums</code> .
Fields	Type the name of the object that contains the field. For example, let's say your <code>Albums</code> object contains a <code>Description</code> field. To find the <code>Description</code> field for albums, type <code>albu</code> , select <code>Albums</code> , and scroll down to <code>Description</code> under Field Permissions.
Apex class access settings	Type <code>apex</code> , then select <code>Apex Class Access</code> .
App and System permissions	Type <code>api</code> , then select <code>API Enabled</code> .
Service providers (available only if Database.com is enabled as an identity provider)	Type <code>serv</code> , then select <code>Service Providers</code> .

If no results appear in a search:

- The permission, object, or setting you're searching for may not be available in the current organization.
- The item you're searching for may not be available for the user license that's associated with the current permission set. For example, a permission set with the Database.com User license doesn't include the “Modify All Data” permission.
- Be sure your search terms have at least three consecutive characters that match the object, setting, or permission name.
- Be sure the search term is spelled correctly.

Assigning Permission Sets

User Permissions Needed	
To view users that are assigned to a permission set:	“View Setup and Configuration”
To assign permission sets:	“Manage Users”

From the user detail page, you can assign permission sets or remove a permission set assignment.

1. Click **Manage Users > Users**.
2. Select a user.
3. In the Permission Set Assignments related list, click **Edit Assignments**.
4. To assign a permission set, select it from the Available Permission Sets box and click **Add**. To remove a permission set assignment, select it from the Enabled Permission Sets box and click **Remove**.



Note: The Permission Set Assignments page shows only permissions sets that match the user's license. For example, if a user's license is Database.com Admin User, you can only assign permission sets with the Database.com Admin User license to that user.

5. Click **Save**.

Managing User Profiles

User Profiles Overview

A profile contains user permissions and access settings that control what users can do within their organization.

Depending on which profile user interface is enabled in your organization, you can:

- [View and edit profiles in the enhanced profile user interface](#)
- [View and edit profiles in the original profile user interface](#)

You can also [use a list view to edit multiple profiles](#).

Profiles control:

- Object permissions that allow users to create, read, edit, and delete records
- Which fields within objects users can view and edit
- Permissions that allow users to manage the system and apps within it
- Which classes users can access
- The hours during which and IP addresses from which users can log in
- Which service providers users can access (if Database.com is enabled as an identity provider)

You can use standard profiles, or create, edit, and delete custom profiles. For standard profiles, only certain settings can be changed.

Each standard or custom profile belongs to exactly one user license type.

Standard Profiles

There are standard profiles in every organization. You can use standard profiles or create, edit, and delete custom profiles.

Profile Name	Available Permissions
Database.com Admin User	Can configure and customize the application.

Profile Name	Available Permissions
Database.com User	Can access Database.com through the API, and can access all standard Chatter people, profiles, groups, and files.



Note: The following profiles appear in the profile list, but can't be assigned to users:

- Chatter Free User
- Chatter Moderator User
- Contract Manager
- Marketing User
- Read Only
- Solution Manager
- Standard User
- System Administrator

Working with Profile Lists

Viewing Profile Lists

User Permissions Needed	
To view profiles, and print profile lists:	“View Setup and Configuration”
To delete profile list views:	“Manage Users”
To delete custom profiles:	<p>“Manage Users”</p> <p>AND</p> <p>“Customize Application”</p>

A profile contains user permissions and access settings that control what users can do within their organization. To view the profiles in your organization, click **Manage Users > Profiles**.

Viewing Enhanced Profile Lists

If enhanced profile list views are enabled for your organization, you can use additional tools to customize, navigate, manage, and print profile lists.

- Show a filtered list of profiles by selecting a view from the drop-down list.
- Delete a view by selecting it from the drop-down list and clicking **Delete**.
- [Create a list view or edit an existing view](#).
- [Create a profile](#).
- Refresh the list view after creating or editing a view by clicking .
- [Edit permissions directly in the list view](#).
- View or edit a profile by clicking its name.
- Delete a custom profile by clicking **Del** next to its name.

Viewing the Basic Profile List

- [Create a profile](#).
- View or edit a profile by clicking its name.
- Delete a custom profile by clicking **Del** next to its name.

Creating and Editing Profile List Views

User Permissions Needed	
To create, edit, and delete profile list views:	“Manage Users”

If [enhanced profile list views](#) are enabled for your organization, you can create profile list views to show a set of profiles with the fields you choose. For example, you could create a list view of all profiles in which “Modify All Data” is enabled.

1. In the Profiles page, click **Create New View**, or select a view and click **Edit**.
2. Enter the view name.
3. Under Specify Filter Criteria, specify the conditions that the list items must match, such as **Modify All Data** equals **True**.
 - a. Type a setting name, or click the lookup icon  to search for and select the setting you want.
 - b. Choose a filter operator.
 - c. Enter the value that you want to match.
 - d. To specify another filter condition, click **Add New**. You can specify up to 25 filter condition rows.

To remove a filter condition row and clear its values, click the remove row icon .

4. Under Select Columns to Display, specify the profile settings that you want to appear as columns in the list view.
 - a. From the Search drop-down list, select the type of setting you want to search for.
 - b. Enter part or all of a word in the setting you want to add and click **Find**.



Note: If the search finds more than 500 values, no results appear. Use the preceding steps to refine your search criteria and show fewer results.

- c. To add or remove columns, select one or more column names and click the **Add** or **Remove** arrow.
- d. Use the **Top**, **Up**, **Down**, and **Bottom** arrows to arrange the columns in the sequence you want.

You can add up to 15 columns in a single list view.

5. Click **Save**, or if you're cloning an existing view, rename it and click **Save As**.

Editing Multiple Profiles with Profile List Views

User Permissions Needed	
To edit multiple profiles from the list view:	“Manage Users”
	AND
	“Customize Application”
	AND
	“Mass Edits from Lists”

If [enhanced profile list views](#) are enabled for your organization, you can change permissions in up to 200 profiles directly from the list view, without accessing individual profile pages. Editable fields display a pencil icon (✎) when you hover over the field, while non-editable fields display a lock icon (🔒). In some cases, such as in standard profiles, the pencil icon appears but the setting is not actually editable.



Caution: Use care when editing profiles with this method. Because profiles affect a user's fundamental access, making mass changes may have a widespread effect on users in your organization.

To change permissions in one or more profiles:

1. Select or [create](#) a list view that includes the profiles and permissions you want to edit.
2. To edit multiple profiles, select the checkbox next to each profile you want to edit. If you select profiles on multiple pages, Database.com remembers which profiles are selected.
3. Double-click the permission you want to edit. For multiple profiles, double-click the permission in any of the selected profiles.
4. In the dialog box that appears, enable or disable the permission. In some cases, changing a permission may also change other permissions. For example, if “Customize Application” and “View Setup and Configuration” are disabled and you enable “Customize Application,” then “View Setup and Configuration” is also enabled. In this case, the dialog box lists the affected permissions.
5. To change multiple profiles, select **All *n* selected records** (where *n* is the number of profiles you selected).
6. Click **Save**.



Note: If you edit multiple profiles, only those profiles that support the permission you are changing will change. For example, if you use inline editing to add “Modify All Data” to multiple profiles, but because of its user license the profile doesn't have “Modify All Data,” the profile won't change.

If any errors occur, an error message appears, listing each profile in error and a description of the error. Click the profile name to open the profile detail page. The profiles you've clicked appear in the error window in gray, strike-through text.



Note: To view the error console, pop-up blockers must be disabled for the Database.com domain. To check if your browser allows pop-up windows, click **My Personal Information > Reminders**, and then click **Preview Reminder Alert**.

Any changes you make are recorded in the [setup audit trail](#).

Cloning Profiles

User Permissions Needed

To create profiles:	“Manage Users”
---------------------	----------------



Tip: If you’re cloning a profile to enable certain permissions or access settings for one or more users, you might be able to enable those permissions or access settings using permission sets. For more information, see [Permission Sets Overview](#).

1. Click **Manage Users > Profiles**.

2. In the Profiles list page, do one of the following:

- Click **New Profile**, then select an existing profile that’s similar to the one you want to create.
- If enhanced profile list views are enabled, click **Clone** next to a profile that’s similar to the one you want to create.
- Click the name of a profile that’s similar to the one you want to create, then in the profile page, click **Clone**.



Note: It’s a good idea to clone only profiles with a Database.com license.

A new profile uses the same [user license](#) as the profile it was cloned from.

3. Enter a profile name.

4. Click **Save**.

Viewing a Profile's Assigned Users

User Permissions Needed

To view users that are assigned to a profile:	“View Setup and Configuration”
To create and edit users:	“Manage Users”

To view all users that are assigned to a profile from the profile overview page, click **Assigned Users** (in the enhanced profile user interface) or **View Users** (in the original profile user interface). From the assigned users page, you can:

- [Create one or multiple users](#)
- [Reset passwords for selected users](#)
- [Edit a user](#)
- [View a user's detail page by clicking the name, alias, or username](#)
- [View or edit a profile by clicking the profile name](#)

Setting Login Restrictions

To help protect your organization's data against unauthorized access, you have several options for setting login restrictions.

Login Hours

For each profile, you can set the hours when users can log in. See:

- [Viewing and Editing Login Hours in the Enhanced Profile User Interface](#) on page 259
- [Viewing and Editing Login Hours in the Original Profile User Interface](#) on page 258

Login IP Address Ranges

For each profile, you can set the IP addresses from which users can log in. See:

- [Viewing and Editing Login IP Ranges in the Enhanced Profile User Interface](#) on page 258
- [Viewing and Editing Login IP Address Ranges in the Original Profile User Interface](#) on page 257

Organization-Wide Trusted IP Address List

For all users, you can set a list of IP address ranges from which they can always log in without receiving a login challenge. See [Restricting Login IP Ranges for Your Organization](#) on page 241.

When users log in to Database.com, either via the user interface, the API, or the Data Loader, Database.com confirms that the login is authorized as follows:

1. Database.com checks whether the user's profile has login hour restrictions. If login hour restrictions are specified for the user's profile, any login outside the specified hours is denied.
2. Database.com then checks whether the user's profile has IP address restrictions. If IP address restrictions are defined for the user's profile, any login from an undesignated IP address is denied, and any login from a specified IP address is allowed.
3. If profile-based IP address restrictions are not set, Database.com checks whether the user is logging in from an IP address they have not used to access Database.com before:
 - If the user's login is from a browser that includes a Database.com cookie, the login is allowed. The browser will have the Database.com cookie if the user has previously used that browser to log in to Database.com, and has not cleared the browser cookies.
 - If the user's login is from an IP address in your organization's trusted IP address list, the login is allowed.
 - If the user's login is from neither a trusted IP address nor a browser with a Database.com cookie, the login is blocked.

Whenever a login is blocked or returns an API login fault, Database.com must verify the user's identity:

- For access via the user interface, the user is prompted to click a **Send Activation Link** button to send an activation email to the address specified on the user's Database.com record. The email instructs the user to copy and paste an activation link into their browser to activate their computer for logging in to Database.com. The activation link included in the email is valid for up to 24 hours from the time the user clicked the **Send Activation Link** button. After 24 hours, the activation link expires, and users must repeat the activation process to log in.



Note: A user will not be asked to activate the first time they log in to Database.com.

- For access via the API or a client, the user must add their security token to the end of their password in order to log in. A security token is an automatically-generated key from Database.com. For example, if a user's password is mypassword, and their security token is XXXXXXXXXXXX, then the user must enter mypasswordXXXXXXXXXX to log in.

Users can obtain their security token by changing their password or resetting their security token via the Database.com user interface. When a user changes their password or resets their security token, Database.com sends a new security token to the email address on the user's Database.com record. The security token is valid until a user resets their security token, changes their password, or has their password reset.



Tip: We recommend that you obtain your security token using the Database.com user interface from a trusted network prior to attempting to access Database.com from a new IP address.

Tips on Setting Login Restrictions

Consider the following when setting login restrictions:

- When a user's password is changed, the user's security token is automatically reset. The user may experience a blocked login until he or she adds the automatically-generated security token to the end of his or her password when logging in to Database.com via the API or a client.
- For more information on API login faults, see the Core Data Types Used in API Calls topic in the [Web Services API Developer's Guide](#).
- If single sign-on is enabled for your organization, API users can't log into Database.com unless their IP address is included on your organization's list of trusted IP addresses or on their profile, if their profile has IP address restrictions set. Furthermore, the single sign-on authority usually handles login lockout policies for users with the "Is Single Sign-On Enabled" permission. However, if the security token is enabled for your organization, then your organization's login lockout settings determine the number of times a user can attempt to log in with an invalid security token before being locked out of Database.com.
- The following events count toward the number of times a user can attempt to log in with an invalid password before being locked out of Database.com, as defined in your organization's login lockout settings:
 - ◊ Each time a user is prompted to click the **Send Activation Link** button
 - ◊ Each time a user incorrectly adds their security token to the end of their password to log into the API or a client

Using the Enhanced Profile User Interface

Enhanced Profile User Interface Overview

The enhanced profile user interface provides a streamlined experience for managing profiles. With it, you can easily navigate, search, and modify settings for a profile.

You can enable the enhanced profile user interface in the [User Interface settings page](#). Your organization can only use one profile user interface at a time.



Note: You can't use the enhanced profile user interface if you use Microsoft® Internet Explorer® 6 or earlier to manage your profiles (unless you've installed the Google Chrome Frame plug-in for Internet Explorer).

Working in the Enhanced Profile User Interface Overview Page

User Permissions Needed	
To view profiles:	"View Setup and Configuration"
To delete profiles and edit profile properties:	"Manage Users"

In the enhanced profile user interface, the profile overview page provides an entry point for all of the settings and permissions for a single profile. To open the profile overview page, click **Manage Users > Profiles** and click the profile you want to view.

From the profile overview page, you can:

- [Search for an object, permission, or setting](#)
- [Clone the profile](#)
- If it's a custom profile that's not assigned to any users, delete the profile by clicking **Delete**
- Change the profile name or description by clicking **Edit Properties**
- [View a list of users who are assigned to the profile](#)
- Click any of the following links to view or edit permissions and settings:

- ◊ Objects and Tabs
- ◊ App Permissions (not available in some profiles)
- ◊ Apex Class Access
- ◊ System Permissions
- ◊ Login Hours
- ◊ Login IP Ranges
- ◊ Service Providers (if Database.com is enabled as an identity provider)

App and System Settings in the Enhanced Profile User Interface

In the enhanced profile user interface, administrators can easily navigate, search, and modify settings for a single profile. Permissions and settings are organized into pages under app and system categories, which reflect the rights users need to administer and use app and system resources.

App Settings

In the enhanced profile user interface, the Apps section of the overview page contains settings that are directly associated with the business processes that the apps enable. The Apps section contains links to these pages:

- Objects and Tabs, which include:
 - ◊ Object Permissions
 - ◊ Field Permissions
- App Permissions (not available in some profiles)
- Apex Class Access

System Settings

Some system functions apply to an organization and not to any single app. For example, login hours and login IP ranges control a user's ability to log in, regardless of which app the user accesses. Other system functions apply to all apps. In some cases, such as with "Modify All Data," a permission applies to all apps, but also includes non-app functions, like the ability to download the Data Loader.

In the enhanced profile user interface, the System section of the overview page contains links to these pages:

- System Permissions
- Login Hours
- Login IP Ranges
- Service Providers (if Database.com is enabled as an identity provider)

Searching in the Enhanced Profile User Interface

User Permissions Needed	
To find permissions and settings in a profile:	"View Setup and Configuration"

On any of the profile pages, type at least three consecutive letters of an object, permission, or setting name in the **Find Settings...** box. The search terms aren't case-sensitive. As you type, suggestions for results that match your search terms appear in a list. Click an item in the list to go to its settings page.

You can search for:

Item	Example
Objects	Type an existing object's name. For example, let's say you have an Albums custom object, type albu, then select Albums.
Fields	Type the name of the object that contains the field. For example, let's say your Albums object contains a Description field. To find the Description field for albums, type albu, select Albums, and scroll down to Description under Field Permissions.
Apex class access settings	Type apex, then select Apex Class Access.
App and system permissions	Type api, then select API Enabled.
Login hours and login IP ranges	Type log, then select Login Hours or Login IP Ranges. Or type ip r, then select Login IP Ranges.
Service providers (available only if Database.com is enabled as an identity provider)	Type serv, then select Service Providers.

If no results appear in a search:

- The permission, object, or setting you're searching for may not be available in the current organization.
- The item you're searching for may not be available for the user license that's associated with the current profile. For example, a profile with the Database.com User license doesn't include the "Modify All Data" permission.
- Be sure your search terms have at least three consecutive characters that match the name of the item you want to find.
- Be sure the search term is spelled correctly.

Viewing and Editing Login Hours in the Enhanced Profile User Interface

User Permissions Needed	
To view login hour settings:	"View Setup and Configuration"
To edit login hour settings:	"Manage Users"

For each profile, you can specify the hours when users can log in.

- Click **Manage Users > Profiles**.
- Select a profile.
- In the profile overview page, click **Login Hours**.
- To change the login hours, click **Edit**.
- Set the days and hours when users with this profile can log in to the organization.

To allow users to log in at any time, click **Clear All Times**. To prohibit users from using the system on a specific day, set the start and end times to the same value.

If a user logs in before the restricted hours, the system ends the user's session when the restricted hours begin.



Note: The first time login hours are set for a profile, the hours are based on the organization's Default Time Zone as specified at **Company Profile > Company Information**. After that, any changes to the organization's Default

Time Zone won't change the time zone for the profile's login hours. As a result, the login hours are always applied at those exact times even if a user is in a different time zone or if the organization's default time zone is changed.

Depending on whether you're viewing or editing login hours, the hours may appear differently. On the Login Hours detail page, hours are shown in your specified time zone. On the Login Hours Edit page, they appear in the organization's original default time zone.

Viewing and Editing Login IP Ranges in the Enhanced Profile User Interface

User Permissions Needed	
To view login IP ranges:	"View Setup and Configuration"
To edit login IP ranges:	"Manage Users"

For each profile, you can view and specify the IP addresses from which users can log in. When you define IP address restrictions for a profile, any login from an undesignated IP address is denied, and any login from a specified IP address is allowed. To view and edit IP address ranges in the enhanced profile user interface:

1. Click **Manage Users > Profiles**.
2. Select a profile.
3. In the profile overview page, click **Login IP Ranges**.
4. Use any of these methods to change login IP address ranges for the profile.
 - Click **Add IP Ranges**. Enter a valid IP address in the IP Start Address and a higher IP address in the IP End Address field. The start and end addresses define the range of allowable IP addresses from which users can log in. To allow logins from a single IP address, enter the same address in both fields. For example, to allow logins from only 125.12.3.0, enter 125.12.3.0 as both the start and end addresses.
 - To change or remove an existing IP address range, click **Edit** or **Delete** for that range.

Both IP addresses in a range must be either IPv4 or IPv6. In ranges, IPv4 addresses exist in the IPv4-mapped IPv6 address space `::ffff:0:0` to `::ffff:ffff:ffff`, where `::ffff:0:0` is `0.0.0.0` and `::ffff:ffff:ffff` is `255.255.255.255`. A range can't include IP addresses inside of the IPv4-mapped IPv6 address space if it also includes IP addresses outside of the IPv4-mapped IPv6 address space. Ranges such as `255.255.255.255` to `::1:0:0:0` or `::` to `::1:0:0:0` are not allowed. You can set up IPv6 addresses in all organizations, but IPv6 is only enabled for login in test database organizations for the Spring '12 release.

Using the Original Profile User Interface

Working with Profiles in the Original Profile Interface

To view a profile, click **Manage Users > Profiles**, then select the profile you want.

On the profile detail page, you can:

- [Edit the profile](#)
- [Create a profile based on this profile](#)
- For custom profiles only, click **Delete** to delete the profile
- [View the users who are assigned to this profile](#)

Editing Profiles in the Original Profile Interface

User Permissions Needed	
To edit profiles:	“Manage Users” AND “Customize Application”

You can edit all settings in a custom profile. In standard profiles, you can edit all settings except name, description, object permissions, field permissions, and user permissions.

1. Click **Manage Users > Profiles**.
2. Select the profile you want to edit.
3. On the profile detail page, click **Edit** to change any of the following settings:
 - For custom profiles only, the name and description
 - For custom profiles only, [administrative and general permissions](#)
 - For custom profiles only, [object permissions](#)



Note: Editing some permissions may automatically cause other permissions to be enabled or disabled. For example, enabling “View All Data” automatically enables “Read” for all objects.



Tip: If enhanced profile list views are enabled for your organization, you can [change permissions for multiple profiles from the list view](#).

You can also view or edit the following settings from the profile detail page:

Setting	Procedure to View or Edit
Access to fields in each object	Under the Field-Level Security section, click View next to an object name.
Login hours	Under the Login Hours section, click Edit .
Login IP address ranges	Under the Login IP Ranges section, click New , or click Edit next to an existing IP range.
Executable Apex classes	Under the Enabled Apex Class Access section, click Edit .

Viewing and Editing Login Hours in the Original Profile User Interface

User Permissions Needed	
To set login hours:	“Manage Users”

For each profile, you can specify the hours when users can log in.

1. Click **Manage Users > Profiles**, and select a profile.
2. Click **Edit** in the Login Hours related list.

3. Set the days and hours when users with this profile can use the system.

To allow users to log in at any time, click **Clear All Times**. To prohibit users from using the system on a specific day, set the start and end times to the same value.

If a user logs in before the restricted hours, the system ends the user's session when the restricted hours begin.

4. Click Save.



Note: The first time login hours are set for a profile, the hours are based on the organization's Default Time Zone as specified at **Company Profile > Company Information**. After that, any changes to the organization's Default Time Zone won't change the time zone for the profile's login hours. As a result, the login hours are always applied at those exact times even if a user is in a different time zone or if the organization's default time zone is changed.

Depending on whether you're viewing or editing login hours, the hours may appear differently. On the profile detail page, hours are shown in your specified time zone. On the Login Hours edit page, they appear in the organization's default time zone.

Viewing and Editing Login IP Address Ranges in the Original Profile User Interface

User Permissions Needed	
To set login IP ranges:	"Manage Users"

You can set the IP addresses from which users with a particular profile can log in. When you define IP address restrictions for a profile, any login from an undesignated IP address is denied, and any login from a specified IP address is allowed. To set IP addresses on profiles:

1. Click **Manage Users > Profiles** and select a profile.
2. Click **New** in the Login IP Ranges related list.
3. Enter a valid IP address in the **IP Start Address** and a higher IP address in the **IP End Address** field.

The start and end addresses define the range of allowable IP addresses from which users can log in. To allow logins from a single IP address, enter the same address in both fields. For example, to allow logins from only 125.12.3.0, enter 125.12.3.0 as both the start and end addresses.

4. Click Save.

Both IP addresses in a range must be either IPv4 or IPv6. In ranges, IPv4 addresses exist in the IPv4-mapped IPv6 address space `::ffff:0:0` to `::ffff:ffff:ffff`, where `::ffff:0:0` is `0.0.0.0` and `::ffff:ffff:ffff` is `255.255.255.255`. A range can't include IP addresses inside of the IPv4-mapped IPv6 address space if it also includes IP addresses outside of the IPv4-mapped IPv6 address space. Ranges such as `255.255.255.255` to `::1:0:0:0` or `::` to `::1:0:0:0` are not allowed. You can set up IPv6 addresses in all organizations, but IPv6 is only enabled for login in test database organizations for the Spring '12 release.

Revoking Permissions and Access

You can use profiles and permission sets to grant access, but not to deny access. Any permission granted from either a profile or permission set is honored. For example, if "Transfer Record" isn't enabled in Jane Smith's profile, but is enabled in two of her permission sets, she can transfer records regardless of whether she owns them. To revoke a permission, you must remove all instances of the permission from the user. You can do this with the following actions—each has possible consequences.

Action	Consequence
Disable a permission or remove an access setting in the profile and any permission sets that are assigned to the user.	The permission or access setting is disabled for all other users assigned to the profile or permission sets.
If a permission or access setting is enabled in the user's profile, assign a different profile to the user.	The user may lose other permissions or access settings associated with the profile or permission sets.
AND	
If the permission or access setting is enabled in any permission sets that are assigned to the user, remove the permission set assignments from the user.	

To resolve the consequence in either case, consider all possible options. For example, you can clone the assigned profile or any assigned permission sets where the permission or access setting is enabled, disable the permission or access setting, and assign the cloned profile or permission sets to the user. Another option is to create a base profile with the least number of permissions and settings that represents the largest number of users possible, then create permission sets that layer additional access.

Permissions Reference

User Permissions

The following table lists all user permissions in alphabetical order. In permission sets and the enhanced profile user interface, these permissions are listed in the App Permissions or System Permissions pages. In the original profile user interface, they are listed under Administrative Permissions and General User Permissions.

You can enable user permissions in [permission sets](#) and [profiles](#).

Permission Name	Description	Functions Controlled	Profiles
API Enabled	Access any Database.com API.	Access any Database.com API	System Administrator Database.com Admin User
API Only User	Access Database.com only through a Database.com API.	Access Database.com only through a Database.com API	Database.com User
Author Apex	Create Apex classes and triggers.	Can modify and deploy Apex classes and triggers , set security on Apex classes, and create email services	System Administrator Database.com Admin User
Bulk API Hard Delete	Delete records in the Bulk API without storing them in the Recycle Bin.	Delete records without storing them in the Recycle Bin, making them eligible for immediate deletion.	None
Chatter Internal User	Use all Chatter features.	Use all Chatter features.	System Administrator Standard User Solution Manager Marketing User

Permission Name	Description	Functions Controlled	Profiles
			Read Only Contract Manager Chatter Moderator User Chatter Free User
Create and Upload Change Sets	Create and upload a change set.	Create a change set, add components to it, and upload it	System Administrator Database.com Admin User
Create AppExchange Packages	Use outbound change sets.	Create and upload change sets.	System Administrator
Customize Application	Customize the organization using App Setup menu options.	Edit messages and custom links; Modify standard picklist values; Create, edit, and delete custom fields; Set field-level security; Manage queues; Create, edit, and delete workflow rules, field updates and outbound messages (outbound messages also requires “Manage Translation” and, if territories are enabled, “Manage Territories” permissions); Create, edit, and delete custom objects;	System Administrator Database.com Admin User
Manage Sharing Calculation Deferral	Suspend and resume sharing calculations.	Suspend and resume group membership and sharing rule calculations.	None
Deploy Change Sets	Deploy change sets.	Deploy change sets sent from another organization	System Administrator Database.com Admin User
Edit Events	Create, edit, and delete events.	Create, edit, and delete events	System Administrator Database.com Admin User Standard User Solution Manager Marketing User

Permission Name	Description	Functions Controlled	Profiles
		Contract Manager	
Edit Read Only Fields	Edit fields that are read only due to page layouts or field-level security.	Edit fields marked as read only (by field-level security or by the page layout) for all other users	System Administrator Database.com Admin User
Edit Tasks	Create, edit, and delete tasks.	Create, edit, and delete tasks	System Administrator Database.com Admin User Standard User Solution Manager Marketing User Contract Manager
Invite Customers to Chatter	Invite customers to Chatter.	Invite customers to Chatter.	System Administrator Standard User Solution Manager Marketing User Read Only Contract Manager Chatter Moderator User Chatter Free User Chatter External User
Is Single Sign-On Enabled	Delegate username and password authentication to a corporate database instead of the Database.com user database.	Username and password authentication is delegated to a corporate database such as Active Directory® or LDAP, instead of the Database.com user database	None
Manage Billing	Purchase additional licenses and features.	Add user licenses; Edit billing and credit card information; Grant Checkout access	System Administrator Database.com Admin User
Manage Chatter Messages	Access all users' messages sent in Chatter.	Access all users' messages sent in Chatter (also requires "API Enabled" or "Api Only User")	None

Permission Name	Description	Functions Controlled	Profiles
Manage Data Integrations	Monitor or abort Bulk API jobs.	Monitor or abort Bulk API jobs; Grant access to Bulk API monitoring pages	System Administrator Database.com Admin User
Manage Public List Views	Create, edit, and delete public list views.	Create, edit, and delete public list views	System Administrator Database.com Admin User Database.com User
Manage Remote Access	Manage, create, edit and delete remote access applications.	Manage, create, edit and delete remote access applications that define integrations of external applications which access Database.com using the OAuth protocol.	System Administrator Database.com Admin User
Manage Users	Create, edit, and deactivate users, and manage security settings, including profiles and roles.	Create, edit, and deactivate users; Define and assign user roles; Define sharing model and sharing rules; View storage use; View login history; View training history; Manage and assign profiles; Set password policies; Set login restrictions	System Administrator Database.com Admin User
Mass Edits from Lists	Edit multiple records simultaneously from a list with inline editing.	Edit two or more records simultaneously from a list with inline editing	System Administrator Database.com Admin User Standard User Solution Manager Marketing User Contract Manager
Moderate Chatter	Deactivate Chatter Free users, assign moderator privileges to Chatter Free users, and remove posts.	Moderate Chatter; Deactivate or re-activate Chatter Free users;	System Administrator Chatter Moderator User

Permission Name	Description	Functions Controlled	Profiles
		Give, or take away, moderator privileges to Chatter Free users; Remove inappropriate posts	
Modify All Data	Create, edit, and delete all organization data, regardless of sharing settings.	Create, edit, and delete all data; Mass delete data; Undelete other users' data;	System Administrator Database.com Admin User
Password Never Expires	Prevent the user's password from expiring.	Prevent password from ever expiring	Database.com Admin User
Reset User Passwords and Unlock Users	Unlock user accounts that are locked, and reset user passwords.	Unlock user accounts that are locked; Reset user passwords	System Administrator Database.com Admin User
Send Outbound Messages	Send outbound messages to an external Web service API.	Send outbound messages to an external Web service API.	System Administrator Database.com Admin User Standard User Database.com User Solution Manager Marketing User Read Only Contract Manager
Tag Manager	Manage the configuration of private and public tags.	Rename, delete, or restore public tags (available only when public tags are enabled)	System Administrator Database.com Admin User
Transfer Record	Change the owner of most records.	Transfer ownership of one or more custom objects that are owned by another user. To transfer records owned by another user, you must also have at least the “Edit” object permission and access to view the records	System Administrator Database.com Admin User
Upload AppExchange Packages	Use outbound change sets.	Create and upload change sets.	System Administrator
View All Data	View all organizational data, regardless of sharing settings.	View all data owned by other users	System Administrator Database.com Admin User

Permission Name	Description	Functions Controlled	Profiles
View Encrypted Data	View the value of encrypted fields in plain text.	View the value of encrypted fields in plain text	None
View Setup and Configuration	View the App Setup and Administrative Settings pages.	View the organization setup details on the Setup pages; View the setup audit trail; Check field accessibility for users	System Administrator Database.com Admin User Standard User Solution Manager Marketing User Read Only Contract Manager

Object Permissions Reference

Object Permissions

Object permissions either respect or override sharing rules and settings. You can enable object permissions in permission sets and custom profiles. The following permissions specify the access that users have to objects.

Permission	Description	Respects or Overrides Sharing?
Read	Users can only view records of this type.	Respects sharing
Create	Users can read and create records.	Respects sharing
Edit	Users can read and update records.	Respects sharing
Delete	Users can read, edit, and delete records.	Respects sharing
View All	Users can view all records associated with this object, regardless of sharing settings.	Overrides sharing
Modify All	Users can read, edit, delete, transfer, and approve all records associated with this object, regardless of sharing settings.	Overrides sharing

"View All" and "Modify All" Permissions Overview

The "View All" and "Modify All" permissions ignore sharing rules and settings, allowing administrators to quickly grant access to records associated with a given object across the organization. "View All" and "Modify All" may be preferable alternatives to the "View All Data" and "Modify All Data" permissions. Be aware of the following distinctions between the permission types.

Permissions	Used for	Users who Need them
View All	Delegation of object permissions	Delegated administrators who need to manage records for specific objects
Modify All		

Permissions	Used for	Users who Need them
View All Data	Managing all data in an organization; for example, Administrators of an entire organization	
Modify All Data	data cleansing, deduplication, mass deletion, mass transferring, and managing record approvals	

“View All” and “Modify All” allow for delegation of object permissions only. To delegate user administration and custom object administration duties, [define delegated administrators](#).

Comparing Security Models

Database.com user security is an intersection of [sharing](#), and [user](#) and [object](#) permissions. In some cases, such as in end-user record level access, it is advantageous to use sharing to provide access to records. In other cases, such as when delegating record administration tasks like transferring records, cleansing data, deduplicating records, and mass deleting records it is advantageous to override sharing and use permissions to provide access to records.

The “Read,” “Create,” “Edit,” and “Delete” permissions respect sharing settings, which control access to data at the record level. The “View All” and “Modify All” permissions override sharing settings for specific objects. Additionally, the “View All Data” and “Modify All Data” permissions override sharing settings for *all* objects.

The following table describes the differences between the security models.

	Permissions that Respect Sharing	Permissions that Override Sharing
Target audience	End-users	Delegated data administrators
Where managed	“Read,” “Create,” “Edit,” and “Delete” object permissions; Sharing settings	“View All” and “Modify All”
Record access levels	Private, Read-Only, Read/Write, Read/Write/Transfer/Full Access	“View All” and “Modify All”
Ability to transfer	Respects sharing settings, which vary by object	Available on all objects with “Modify All”
Object support	Available on all objects	Available on objects via object permissions
Group access levels determined by	Roles, Roles and Subordinates, Queues, and Public Groups	Profile or permission sets
Ability to manually share records	Available to the record owner and any user above the record owner in the role hierarchy	Available on all objects with “Modify All”

Field Permissions

Field permissions specify the access level for each field in an object. In permission sets and the enhanced profile user interface, the setting labels differ from those in the original profile user interface and in field-level security pages for customizing fields.

Access Level	Enabled Settings in Permission Sets and Enhanced Profile User Interface	Enabled Settings in Original Profile and Field-Level Security Interfaces
Users can read and edit the field.	Read and Edit	Visible
Users can read but not edit the field.	Read	Visible and Read-Only
Users can't read or edit the field.	None	None

Managing Tags

Tags Settings

User Permissions Needed
To modify tag settings: "Customize Application"

Administrators set up and manage personal and public tags by [Enabling tags](#) for custom objects and [Deleting personal tags](#) for deactivated users.

Enabling Tags

User Permissions Needed
To modify tag settings: "Customize Application"

1. Click **Customize > Tags > Tag Settings**.
2. Select Enable Personal Tags and Enable Public Tags to allow users to add personal and public tags to records. Deselect both options to disable tags.
3. Specify which objects should display tags in a tag section at the top of record detail pages. The tag section is the only way that a user can add tags to a record.
4. Click **Save**.

Deleting Personal Tags for Deactivated Users

User Permissions Needed
To delete personal tags for deactivated users: "Customize Application"

Your organization can have a maximum of 5,000,000 personal and public tags applied to records across all users. If your organization is approaching this limit, you can delete personal tags for deactivated users.

1. Click **Customize > Tags > Personal Tag Cleanup**.
2. Select one or more deactivated users and click **Delete**.

You can't restore personal tags after you delete them.

Working with Queues

Queues Overview

User Permissions Needed	
To create or change queues: To create or change queues created by other users:	“Customize Application” AND “Manage Public List Views” “Customize Application” AND “Manage Public List Views” and “Manage Users”

When a custom object record is created, manually assign it to a queue so that the users assigned to the queue can access and take ownership of it. Any queue member or users above them in the role hierarchy can take ownership of records in a queue.

Managing Queues

User Permissions Needed	
To create or change queues: To create or change queues created by other users:	“Customize Application” AND “Manage Public List Views” “Customize Application” AND “Manage Public List Views” and “Manage Users”

To view and manage queues, click **Manage Users** > **Queues**.

- To create a queue, click **New**.
- To edit a queue, click **Edit** next to the queue name.
- To delete a queue, click **Del** next to the queue name.
- To view details about a queue, including the queue's members, click the queue's name.

Creating Queues

User Permissions Needed	
To create or change queues:	“Customize Application” AND “Manage Public List Views”
To create or change queues created by other users:	“Customize Application” AND “Manage Public List Views” and “Manage Users”

1. Click **Manage Users > Queues**.

2. Click **New**.

3. Enter the Label and Queue Name. The Label is the queue label as it appears on the user interface. The Queue Name is a unique name used by the API and managed packages, and can only contain alphanumeric characters and underscores.

4. Choose the objects available to the queue.

5. Choose queue members.

You can select individual users, roles, or public groups. Only queue members and users above them in the role hierarchy can take ownership of records in the queue, depending on your organization's sharing settings.

6. Click **Save**.

Tips on Creating Queues

- After you have defined a queue for an object, you can set up a workflow rule that automatically reassigns ownership of object records to the queue via a field update on the record owner field.

Viewing and Editing Queues

User Permissions Needed	
To create or change queues:	“Customize Application” AND “Manage Public List Views”
To create or change queues created by other users:	“Customize Application” AND “Manage Public List Views” and “Manage Users”

To view or edit a queue:

1. Click **Manage Users > Queues** and select the queue name.

2. Click:

- Edit** to make changes to the queue.

- **Delete** to remove the queue.
You can't delete a queue if it's in an assignment rule or has records in it.
- **View All Users** to view the queue's members plus any users who have the same access as queue members because they're higher in your organization's role hierarchy.
You can click **View Queue Members** to return to the Queue Members related list.

3. If you edited the queue, click **Save**.

Viewing the Queue Membership Related List

User Permissions Needed	
To view user details:	"View Setup and Configuration"

To view the queues a user is a member of, click **Manage Users > Users** and select the user. In the Queue Membership related list, you can:

- Click **New Queue** to create a queue.
- Click a queue name to view its details.

Monitoring User Events

User Permissions Needed	
To monitor logins and view training history:	"Manage Users"

You can monitor user logins and trainings to determine who is using Database.com and how often and which users have taken training classes from salesforce.com.

If your organization is enabled for Single Sign-On using delegated authentication and has built a Single Sign-On solution, you can view the most recent Single Sign-On login errors for your organization.

Managing Security

Security Overview

Database.com is built with security as the foundation for the entire service. This foundation includes both protection for your data and applications, and the ability to implement your own security scheme to reflect the structure and needs of your organization.

The security features of Database.com provide both strength and flexibility. However, protecting your data is a joint responsibility between you and salesforce.com. The security features in Database.com enable you to empower your users to do their jobs efficiently, while also limiting exposure of data to the users that need to act upon it. Implement security controls that you think

are appropriate for the sensitivity of your data. Your data is protected from unauthorized access from outside your company, and you should also safeguard it from inappropriate usage by your users.

See the following topics to get more information about the various security components in Database.com:

- [Security Infrastructure](#) on page 237
- [Trust and Salesforce.com](#) on page 236
- [User Security Overview](#) on page 255
- [About Passwords](#) on page 198
- [User Authentication](#) on page 255
- [Identity Providers](#) on page 304
- [Network-Based Security](#) on page 257
- [CAPTCHA Security for Data Exports](#) on page 316
- [Session Security](#) on page 239
- [Securing Data Access](#) on page 237
- [Auditing](#) on page 239

Trust and Salesforce.com

Trust starts with transparency and that is why salesforce.com displays real-time information on system performance and security on the trust site at <http://trust.database.com>. On this site, you can find live data on system performance, current and recent phishing and malware attempts, and tips on best security practices for your organization.

The Security tab on the trust site includes valuable information that can help you to safeguard your company's data. In particular, phishing and malware are Internet scams on the rise.

Phishing is a social engineering technique that attempts to acquire sensitive information such as usernames, passwords and credit card details, by masquerading as a trustworthy entity in an electronic communication. Phishers often direct users to enter details at a fake website whose URL and look-and-feel are almost identical to the legitimate one. As the salesforce.com community grows, it has become an increasingly appealing target for phishers. You will never get an email or a phone call from a salesforce.com employee asking you to reveal a password, so you should refuse to reveal it to anyone. You can report any suspicious activities by clicking the **Report a Suspicious Email** link under the **Trust** tab at <http://trust.database.com>.

Malware is software designed to infiltrate or damage a computer system without the owner's informed consent. It is a general term used to cover a variety of forms of hostile, intrusive, or annoying software, and it includes computer viruses and spyware.

What Salesforce.com is Doing

Customer security is the foundation of customer success, so salesforce.com will continue to implement the best possible practices and technologies in this area. Recent and ongoing actions include:

- Actively monitoring and analyzing logs to enable proactive alerts to customers who have been affected.
- Collaborating with leading security vendors and experts on specific threats.
- Executing swift strategies to remove or disable fraudulent sites (often within an hour of detection).
- Reinforcing security education and tightening access policies within salesforce.com.
- Evaluating and developing new technologies both for our customers and for deployment within our infrastructure.

What Salesforce.com Recommends You Do

Salesforce.com is committed to setting the standards in software-as-a-service as an effective partner in customer security. So, in addition to internal efforts, salesforce.com strongly recommends that customers implement the following changes to enhance security:

- Modify your Database.com implementation to activate IP range restrictions. This will allow users to access Database.com only from your corporate network or VPN, thus providing a second factor of authentication. For more information, see [Setting Session Security](#) on page 240 and [Restricting Login IP Ranges for Your Organization](#) on page 241.
- Educate your employees not to open suspect emails and to be vigilant in guarding against phishing attempts.
- Use security solutions from leading vendors such as Symantec to deploy spam filtering and malware protection.
- Designate a security contact within your organization so that salesforce.com can more effectively communicate with you. Contact your salesforce.com representative with this information.
- Consider using two-factor authentication techniques, such as RSA tokens, to restrict access to your network.

Salesforce.com has a Security Incident Response Team to respond to any security issues. To report a security incident with Database.com, contact security@salesforce.com. Describe the incident in detail, and the team will respond promptly.

Security Infrastructure

One of the core features of a multi-tenant platform is the use of a single pool of computing resources to service the needs of many different customers. Database.com protects your organization's data from all other customer organizations by using a unique organization identifier, which is associated with each user's session. Once you log in to your organization, your subsequent requests are associated with your organization, using this identifier.

Database.com utilizes some of the most advanced technology for Internet security available today. When you access the application using a Database.com-supported browser, Secure Socket Layer (SSL) technology protects your information using both server authentication and data encryption, ensuring that your data is safe, secure, and available only to registered users in your organization.

In addition, Database.com is hosted in a secure server environment that uses a firewall and other advanced technology to prevent interference or access from outside intruders.

Managing System Security

Securing Data Access

Choosing the data set that each user or group of users can see is one of the key decisions that affects data security. You need to find a balance between limiting access to data, thereby limiting risk of stolen or misused data, versus the convenience of data access for your users.

To enable users to do their job without exposing data that they do not need to see, Database.com provides a flexible, layered sharing design that allows you to expose different data sets to different sets of users.

- To specify the objects that users can access, you can assign permission sets and profiles.
- To specify the fields that users can access, you can use field-level security.
- To specify the individual records that users can view and edit, you can set your organization-wide sharing settings, define a role hierarchy, and create sharing rules.



Tip: When implementing security and sharing rules for your organization, make a table of the various types of users in your organization. In the table, specify the level of access to data that each type of user needs for each object and for fields and records within the object. You can refer to this table as you set up your security model.

The following describes these security and sharing settings:

Object-Level Security (Permission Sets and Profiles)

Object-level security provides the bluntest way to control data. Using object-level security you can prevent a user from seeing, creating, editing, or deleting any instance of a particular type of object. Object-level security lets you hide whole objects from particular users, so that they don't even know that type of data exists.

You specify object-level security settings in [permission sets](#) and [profiles](#). *Permission sets* and *profiles* are collections of settings and permissions that determine what a user can do in the application, similar to a group in a Windows network, where all of the members of the group have the same folder permissions and access to the same software.

Profiles are typically defined by a user's job function (for example, system administrator or sales representative). A profile can be assigned to many users, but a user can be assigned to only one profile. You can use permission sets to grant additional permissions and access settings to users. It's easy to manage users' permissions and access with permission sets, because you can assign multiple permission sets to a single user.

Field-Level Security (Permission Sets and Profiles)

In some cases, you may want users to have access to an object, but limit their access to individual fields in that object. Field-level security controls whether a user can see, edit, and delete the value for a particular field on an object. It lets you protect sensitive fields without having to hide the whole object from users. Field-level security is also controlled in permission sets and profiles.

Record-Level Security (Sharing)

After setting object- and field-level access permissions, you may want to configure access settings for the actual records themselves. Record-level security lets you give users access to some object records, but not others. Every record is owned by a user or a queue. The owner has full access to the record. In a hierarchy, users higher in the hierarchy always have the same access to users below them in the hierarchy. This access applies to records owned by users, as well as records shared with them.

To specify record-level security, set your organization-wide sharing settings, define a hierarchy, and create sharing rules.

- [Organization-wide sharing settings](#)—The first step in record-level security is to determine the organization-wide sharing settings for each object. Organization-wide sharing settings specify the default level of access users have to each others' records.

You use organization-wide sharing settings to lock down your data to the most restrictive level, and then use the other record-level security and sharing tools to selectively give access to other users. For example, let's say users have object-level permissions to read and edit albums, and the organization-wide sharing setting is Read-Only. By default, those users can read all album records, but can't edit any unless they own the record or are granted additional permissions.

- [Role hierarchy](#)—Once you've specified organization-wide sharing settings, the first way you can give wider access to records is with a role hierarchy. Similar to an organization chart, a role hierarchy represents a level of data access that a user or group of users needs. The role hierarchy ensures that users higher in the hierarchy always have access to the same data as people lower in their hierarchy, regardless of the organization-wide default settings. Role hierarchies don't have to match your organization chart exactly. Instead, each role in the hierarchy should represent a level of data access that a user or group of users needs.



Note: Although it's easy to confuse permission sets and profiles with roles, they control two very different things. Permission sets and profiles control a user's object and field access permissions. Roles primarily control a user's record-level access through role hierarchy and sharing rules.

- **Sharing rules**—Sharing rules let you make automatic exceptions to organization-wide sharing settings for particular sets of users, to give them access to records they don't own or can't normally see. Sharing rules, like role hierarchies, are only used to give additional users access to records—they can't be stricter than your organization-wide default settings.
- **Manual sharing**—Sometimes it's impossible to define a consistent group of users who need access to a particular set of records. In those situations, record owners can use manual sharing to give read and edit permissions to users who would not have access to the record any other way. Although manual sharing isn't automated like organization-wide sharing settings, role hierarchies, or sharing rules, it gives record owners the flexibility to share particular records with users that need to see them.
- **Apex managed sharing**—If sharing rules and manual sharing don't give you the control you need, you can use Apex managed sharing. Apex managed sharing allows developers to programmatically share custom objects. When you use Apex managed sharing to share a custom object, only users with the "Modify All Data" permission can add or change the sharing on the custom object's record, and the sharing access is maintained across record owner changes.

Auditing

Auditing features do not secure your organization by themselves, but these features provide information about usage of the system, which can be critical in diagnosing potential or real security issues. It is important that someone in your organization perform regular audits to detect potential abuse. The other security features provided by Database.com are preventative. To verify that your system is actually secure, you should perform audits to monitor for unexpected changes or usage trends.

Auditing features include:

Record Modification Fields

All objects include fields to store the name of the user who created the record and who last modified the record. This provides some basic auditing information.

Login History

You can review a list of successful and failed login attempts to your organization for the past six months. See [Monitoring Login History](#) on page 65.

Field History Tracking

You can also enable auditing for individual fields, which will automatically track any changes in the values of selected fields. See [Tracking Field History](#) on page 242.

Setup Audit Trail

Administrators can also view a Setup Audit Trail, which logs when modifications are made to your organization's configuration. See [Monitoring Setup Changes](#) on page 311.

Managing Session Security

Session Security

After logging in, a user establishes a session with Database.com. Use session security to limit exposure to your network when a user leaves their computer unattended while still logged on. It also limits the risk of internal attacks, such as when one employee tries to use another employee's session.

You can control the session expiration time window for user logins. Session expiration allows you to select a timeout for user sessions. The default session timeout is two hours of inactivity. When the session timeout is reached, users are prompted with a dialog that allows them to log out or continue working. If they do not respond to this prompt, they are automatically logged out.

By default, Database.com uses SSL (secure sockets layer) and requires secure connections (HTTPS) for all communication. The **Require secure connections (HTTPS)** setting determines whether SSL (HTTPS) is required for access to Database.com. If you disable this setting and change the URL from `https://` to `http://`, you can still access the application. However, you should require all sessions to use SSL for added security. See [Setting Session Security](#) on page 240.

Setting Session Security

User Permissions Needed	
To set session security:	“Customize Application”

You can modify session security settings to specify connection type, timeout settings, and more.

1. Click **Security Controls > Session Settings**.
2. Customize the session security settings.

Field	Description
Timeout value	<p>Length of time after which the system prompts users who have been inactive to log out or continue working. Select a value between 30 minutes and 12 hours. Choose a shorter timeout period if your organization has sensitive information and you want to enforce stricter security.</p> <p> Note: The last active session time value isn't updated until halfway through the timeout period. That is, if you have a 30 minute timeout, the system won't check for activity until 15 minutes have passed. For example, assume you have a 30 minute timeout value. If you update a record after 10 minutes, the last active session time value won't be updated because there was no activity after 15 minutes. You'll be logged out in 20 more minutes (30 minutes total) because the last active session time wasn't updated. Suppose you update a record after 20 minutes. That's five minutes after the last active session time is checked, so your timeout resets and you have another 30 minutes before being logged out, for a total of 50 minutes.</p>
Disable session timeout warning popup	Determines whether the system prompts users with a timeout warning message after any length of inactivity. Select this option to provide extra security.
Require secure connections (HTTPS)	<p>Determines whether logins and access to Database.com are required to use HTTPS.</p> <p>This option is enabled by default. You should require HTTPS connections for enhanced security. Once this preference is set to require HTTPS, you can't manually change it. To change to HTTP, contact your salesforce.com representative.</p> <p> Note: The Resetting Passwords on page 201 page can only be accessed using HTTPS.</p>

Field	Description
Force relogin after Login-As-User	Determines whether an administrator that is logged in as another user is returned to their previous session after logging out as the secondary user. For more information, see Logging In as Another User on page 202. If the option is enabled, an administrator must log in again to continue using Database.com after logging out as the user; otherwise, the administrator is returned to their original session after logging out as the user.
Enable caching and password autocomplete on login page	Allows the user's browser to store usernames. If enabled, after an initial log in, usernames are auto-filled into the User Name field on the login page. This preference is selected by default and caching and autocomplete are enabled.

- Click **Save**.

Restricting Login IP Ranges for Your Organization

User Permissions Needed	
To view network access:	“Login Challenge Enabled”
To change network access:	“Manage Users”

To help protect your organization's data from unauthorized access, you can specify a list of IP addresses from which users can always log in without receiving a login challenge:

- Click **Security Controls > Network Access**.
- Click **New**.
- Enter a valid IP address in the `Start IP Address` field and a higher IP address in the `End IP Address` field.

The start and end addresses define the range of allowable IP addresses from which users can log in. If you want to allow logins from a single IP address, enter the same address in both fields. For example, to allow logins from only 125.12.3.0, enter 125.12.3.0 as both the start and end addresses.

The start and end IP addresses in an IPv4 range must include no more than 33,554,432 addresses (2^{25} , a /7 CIDR block). For example, the following ranges are valid:

- 0.0.0.0 to 1.255.255.255
- 132.0.0.0 to 132.255.255.255
- 132.0.0.0 to 133.255.255.255

However, ranges like 0.0.0.0 to 2.255.255.255 or 132.0.0.0 to 134.0.0.0 are too large.

The start and end IP addresses in an IPv6 range must include no more than 79,228,162,514,264,337,593,543,950,336 addresses (2^{96} , a /32 CIDR block). For example, the following range is valid: 2001:8000:: to 2001:8000:ffff:ffff:ffff:ffff:ffff:ffff. However, ranges like :: to ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff or 2001:8000:: to 2001:8001:: are too large.

4. Click **Save.**



Note: Both IP addresses in a range must be either IPv4 or IPv6. In ranges, IPv4 addresses exist in the IPv4-mapped IPv6 address space `::ffff:0:0` to `::ffff:ffff:ffff`, where `::ffff:0:0` is `0.0.0.0` and `::ffff:ffff:ffff` is `255.255.255.255`. A range can't include IP addresses inside of the IPv4-mapped IPv6 address space if it also includes IP addresses outside of the IPv4-mapped IPv6 address space. Ranges such as `255.255.255.255` to `::1:0:0:0` or `::` to `::1:0:0:0` are not allowed. You can set up IPv6 addresses in all organizations, but IPv6 is only enabled for login in test database organizations for the Spring '12 release.

Tracking Field History

User Permissions Needed	
To set up which fields are tracked:	"Customize Application"

You can select certain custom fields to track the history of related list of custom objects. Modifying any of these standard or custom fields adds a new entry to the History related list. All entries include the date, time, nature of the change, and who made the change. History data does not count against your organization's storage limit. Note that not all field types are available for history tracking.

For more information on tracking field history, see the following:

- [Tracking Field History for Custom Objects](#)
- [History Tracking Implementation Tips](#)

To set up field history tracking:

1. Click **Customize** and select the object you want to configure.
2. Click **Fields**.
3. Click **Set History Tracking**.
 - When you choose the fields you want to track, Database.com begins tracking history from that date and time forward. Changes made before that date and time are not included.
4. Choose the fields you want tracked.
5. Click **Save**.

Tracking Field History for Custom Objects

To track field history for custom objects:

1. Click **Create > Objects** and click **Edit** next to the name of the custom object.
2. Select the **Track Field History** checkbox. Deselect the checkbox if you do not want to track any changes. If you deselect the checkbox, the History related list is automatically removed from the custom object's page layouts.
3. Click **Save**.
4. Select the name of the custom object.
5. Click **Set History Tracking** in the Custom Fields & Relationships section. This section allows you to set a custom object's history tracking for both standard and custom fields.

When you choose the fields you want to track, Database.com begins tracking history from that date and time forward. Changes made before that date and time are not included.

If you deselected the **Track Field History** checkbox, the **Set History Tracking** button does not display.

6. Choose the fields you want tracked.
7. Click **Save**.

History Tracking Implementation Tips

Administration

- You can select a combination of up to 20 custom fields per object.
- You cannot track the following fields:
 - ◊ History of formula, roll-up summary, or auto-number
 - ◊ Created By and Last Modified By
- Field history is stored for 18 months.
- To archive field history, you can:
 - ◊ Schedule a regular export of `FieldHistory` data
 - ◊ Run a query using the [Web services API](#) and save your results
- If you disable field history tracking on an object, you can still report on its history data up to the date and time you disabled tracking.
- You cannot disable field history tracking for an object if a field on the object is referenced in an Apex script. For more information, see [Apex Code Overview](#) on page 379.
- If the parent record in a lookup relationship is deleted, the field history tracking for the child record does not record the deletion.

Customization

- You cannot customize the History related list because it does not store data. The History related list links to data stored elsewhere.
- When you delete a custom field, all of the field history data is deleted and changes are no longer tracked.
- If you disable field history tracking on a custom object, then you cannot report on its field history.

Management

- Changes to fields with more than 255 characters are tracked as edited, and their old and new values are not recorded. For example, changes to long text area fields are tracked as edited.
- Changes to date fields, number fields, and standard field labels are shown in the locale of the user viewing the History related list. For example, a date change to August 8, 2005 shows as 8/5/2005 for a user with the English (United States) locale and as 5/8/2005 for a user with the English (United Kingdom) locale.
- [Field updates](#) are tracked in the History related list if you have set history tracking on those fields.

Delegating Administrative Duties

Delegating Administrative Duties

User Permissions Needed	
To delegate administration:	“Customize Application”
To be a delegated administrator:	“View Setup and Configuration”

Use delegated administration to assign limited administrative privileges to selected non-administrator users in your organization.

Delegated administrators can perform the following tasks:

- Creating and editing users and resetting passwords for users in specified roles and all subordinate roles
- Assigning users to specified profiles
- Logging in as a user who has [granted login access](#) to their administrator
- Managing custom objects created by an administrator

For example, you may want to allow the manager of the Customer Support team to create and edit users in the Support Manager role and all subordinate roles. This allows the administrator to focus on tasks other than managing users for every department that uses Database.com.

To create delegated groups, click **Security Controls > Delegated Administration**, then click **New**.

To manage your delegated groups:

- Click **Edit** next to a group to modify it.
- Select a group and click **Delete** to remove it.
- Select a group and click **Remove** next to the user in the Delegated Administrators related list to remove a user from that delegated group.



Note: To delegate administration of particular objects, use [object permissions](#), such as “View All” and “Modify All.”

Defining Delegated Administrators

User Permissions Needed	
To delegate administration:	“Customize Application”
To be a delegated administrator:	“View Setup and Configuration”

Define delegated administration groups to specify groups of users who you want to have the same administrative privileges. These groups are not related to public groups used for sharing.

1. Click **Security Controls > Delegated Administration**.
2. Click **New**.
3. Enter a group name.
4. Select **Enable Group for Login Access** if you want to allow delegated administrators in this group to log in as users who have granted login access to their administrators and are in the roles selected for the delegated administrator group. To find out how users can grant login access to their administrators, see [Granting Login Access](#) on page 79.
5. Click **Save**.
6. Click **Add** in the Delegated Administrators related list to specify the users in this delegated group.
7. Use the magnifying glass lookup icon to find and add users to the group. The users must have the “View Setup and Configuration” permission.
8. Click **Save**.
9. See [Delegating User Administration](#) on page 245 and [Delegating Custom Object Administration](#) on page 245 to specify what tasks these users can perform.

Delegating User Administration

User Permissions Needed	
To delegate administration:	“Customize Application”
To be a delegated administrator:	“View Setup and Configuration”

Enable delegated administrators to manage users in specified roles and all subordinate roles, assign specified profiles to those users, and log in as users who have [granted login access](#) to administrators.

1. Click **Security Controls > Delegated Administration**.
2. Select the name of an existing delegated administration group.
3. Click **Add** in the User Administration related list.
4. Use the magnifying glass lookup icon to find and add roles. Delegated administrators can create and edit users in these roles and all subordinated roles.
5. Click **Save**.
6. Click **Add** in the Assignable Profiles related list.
7. Use the magnifying glass lookup icon to find and add profiles. Delegated administrators can assign these profiles to the users they create and edit. Note that profiles with the “Modify All Data” permission cannot be assigned by delegated administrators.



Note: If a user is a member of more than one delegated administration group, be aware that he or she can assign any of the assignable profiles to any of the users in roles he or she can manage.

8. Click **Save**.
9. See [Delegating Custom Object Administration](#) on page 245 to specify what custom objects the delegated administrators can manage.

To remove roles or profiles from the list of items the delegated administrators can use, click **Remove** next to the role or profile.

Delegating Custom Object Administration

User Permissions Needed	
To delegate administration:	“Customize Application”
To be a delegated administrator:	“View Setup and Configuration”

Enable delegated administrators to manage custom objects that have been created by an administrator.

1. Click **Security Controls > Delegated Administration**.
2. Select the name of an existing delegated administration group.
3. From the detail page of the delegated administration group, click **Add** in the Custom Object Administration related list.
4. Use the magnifying glass lookup icon to find and add custom objects. Delegated administrators can customize nearly every aspect of a custom object, including creating a custom tab for it.
5. Click **Save**. Click **Save & More** to add additional custom objects.

To remove a custom object from the list of items the delegated administrators can manage, click **Remove** next to the custom object.

Notes on Delegated Administration of Custom Objects

- Delegated administrators can customize nearly every aspect of the custom object. However, they cannot create or modify relationships on the object or set organization-wide sharing defaults.
- Delegated administrators need to have access to custom objects if they need to access the merge fields on those objects from formulas.

Concurrent Usage Limits

To ensure that resources are available for all Database.com users, limits are placed on the number of long-running Web requests that one organization can send at the same time. This kind of limit is called a maximum *concurrent Web requests limit*.

Database.com monitors the number of concurrent requests issued by all users logged in to your organization, and compares that number against the maximum limit. In this way, the number of concurrent requests is kept below the maximum limit. The limit ensures that resources are available uniformly to all organizations and prevents deliberate or accidental over-consumption by any one organization.

If too many requests are issued by users in your organization, you may have to wait until one of them has finished before you can perform your task. For example, assume that your custom reporting application has 100,000 users. At 9:00 AM, each user requests a report that contains 200,000 records. Your application starts to run the report for all users until the maximum number of concurrent requests has been met. At that point, Database.com refuses to take any additional requests until some of the reports have completed.

Similar limits are placed on requests issued from the API. For more information, see [Database.com Limits](#) in *Getting Started with Database.com*.

Managing Sharing

About Organization-Wide Sharing Defaults

Administrators can use organization-wide sharing settings to define the default sharing settings for an organization.

Organization-wide sharing settings specify the default level of access to records and can be set separately for custom objects. For most objects, organization-wide sharing settings can be set to Private, Public Read Only, or Public Read/Write.

In environments where the organization-wide sharing setting for an object is Private or Public Read Only, an administrator can grant users additional access to records by setting up a role hierarchy or defining sharing rules. However, sharing rules can only be used to grant additional access—they cannot be used to restrict access to records beyond what was originally specified with the organization-wide sharing defaults.

By default, Database.com uses hierarchies to automatically grant access of records to users above the record owner in the hierarchy.

Use the **Grant Access Using Hierarchies** checkbox to disable access to records to users above the record owner in the hierarchy for custom objects. If you deselect this checkbox for a custom object, only the record owner and users granted access by the organization-wide defaults receive access to the records.

Default Organization-Wide Sharing Settings

The default organization-wide sharing setting for custom objects is Public Read/Write.

Sharing Considerations

Your organization's sharing model gives users access to records they do not own. The sharing model is a complex relationship between role hierarchies, user permissions, sharing rules, and exceptions for certain situations. Review the following notes before setting your sharing model:

Exceptions to Role Hierarchy-based Sharing

Users can always view and edit all data owned by or shared with users below them in the role hierarchy. Exceptions to this include:

- An option on your organization-wide default allows you to ignore the hierarchies when determining access to data. For more information on this setting, see “Controlling Access Using Hierarchies” in the online help.
- Users above a record owner in the role hierarchy can only view or edit the record owner’s records if they have the “Read” or “Edit” object permission for the type of record

Deleting Records

- The ability to delete individual records is controlled by administrators, the record owner, users in a role hierarchy above the record owner, and any user that has been granted “Full Access.”

User Permissions and Object-Level Permissions

While your sharing model controls visibility to records, user permissions and object-level permissions control what users can do to those records.

- Regardless of the sharing settings, users must have the appropriate object-level permissions. For example, if you share an album, those users can only see the album if they have the “Read” permission on albums. Likewise, users who have the “Edit” permission on albums may still not be able to edit albums they do not own if they are working in a Private sharing model.
- Administrators, and users with the “View All Data” or “Modify All Data” permissions, have access to view or edit all data.

Apex Sharing

You can't change the organization-wide default settings from private to public for a custom object if an [Apex](#) script uses the sharing entries associated with that object. For example, if a script retrieves the users and groups who have sharing access on a custom object `Invoice__c` (represented as `Invoice__share` in the code), you can't change the object's organization-wide sharing setting from private to public.

Viewing Sharing Overrides

User Permissions Needed	
To view sharing overrides:	“View Setup and Configuration”

When you select an object in the Sharing Settings page, the page includes a Sharing Overrides related list, which shows any profiles that ignore sharing settings for that object.

To view the Sharing Overrides list, click **Security Controls > Sharing Settings**, then select an object from the Manage Sharing Settings For list.

For each profile, the list specifies the permissions that allow it to override sharing settings. The “View All Data” and “Modify All Data” permissions override sharing settings for all objects in the organization, while the object permissions “View All” and “Modify All” override sharing settings for the named object.



Note: The Sharing Overrides list doesn't show permissions granted through permission sets, which may also override sharing settings for an object.

To override sharing settings for specific objects, you can create or edit permission sets or profiles and enable the “View All” and “Modify All” object permissions. These permissions provide access to all records associated with an object across the organization, regardless of the sharing settings. Before setting these permissions, compare the different ways to control data access.

Setting Your Organization-Wide Sharing Defaults

User Permissions Needed	
To set default sharing access:	“Manage Users” AND “Customize Application”

1. Click **Security Controls > Sharing Settings**.
2. Click **Edit** in the Organization-Wide Defaults area.
3. For each object, select the default access you want.
4. To disable automatic access using your hierarchies, deselect **Grant Access Using Hierarchies** for any custom object that does not have a default access of Controlled by Parent.



Note: If **Grant Access Using Hierarchies** is deselected, users that are higher in the role hierarchy don't receive automatic access. However, some users—such as those with the “View All” and “Modify All” **object permissions** and the “**View All Data**” and “**Modify All Data**” system permissions—can still access records they don't own.

Limitations

You can't change the organization-wide sharing default setting for some objects:

- When a custom object is on the detail side of a master-detail relationship with a standard object, its organization-wide default is set to Controlled by Parent and it is not editable.

You can't change the organization-wide default settings from private to public for a custom object if an **Apex** script uses the sharing entries associated with that object. For example, if a script retrieves the users and groups who have sharing access on a custom object `Invoice__c` (represented as `Invoice__share` in the code), you can't change the object's organization-wide sharing setting from private to public.

Managing Sharing Rules

Managing the Sharing Settings

User Permissions Needed	
To set default sharing access:	“Manage Users”
	AND
	“Customize Application”

Using Database.com, you can control access to data at many different levels. For example, you can control the access your users have to objects with object permissions. Within objects, you can control the access users have to fields using field-level security. To control access to data at the record level, use the sharing settings described below.

To view your sharing settings, click **Security Controls > Sharing Settings**. You can either view all lists at once, or you can use the **Manage sharing settings** for drop-down list at the top of the page to view only organization-wide defaults and sharing rules for a selected object.

Organization-Wide Defaults

Your [organization-wide default sharing settings](#) give you a baseline level of access for each object and enable you to extend that level of access using hierarchies or sharing rules. For example, you can set the organization-wide default for a custom object to Private if you only want users to view and edit the records they own of that custom object. Then, you can create sharing rules for the custom object to extend access of the object's records to particular users or groups.

Sharing Rules

[Sharing rules](#) represent the exceptions to your organization-wide default settings. If you have organization-wide sharing defaults of Public Read Only or Private, you can define rules that give additional users access to records they do not own. You can create sharing rules based on record owner or field values in the record.



Tip: Sometimes it's impossible to define a consistent group of users who need access to a particular set of records. In those situations, record owners can use manual sharing to give read and edit permissions to users who would not have access to the record any other way. Although manual sharing isn't automated like organization-wide sharing settings, role hierarchies, or sharing rules, it gives record owners the flexibility to share particular records with users that need to see them.

Apex Managed Sharing

Apex managed sharing allows developers to programmatically share custom objects. When you use Apex managed sharing to share a custom object, only users with the “Modify All Data” permission can add or change the sharing on the custom object's record, and the sharing access is maintained across record owner changes.

For more information on Apex managed sharing, see the [Database.com Apex Code Developer's Guide](#).

Other Methods for Allowing Access to Records

In addition to sharing settings, there are a few other ways to allow multiple users access to given records:

Queues

When a custom object record is created, manually assign it to a queue so that the users assigned to the queue can access and take ownership of it. Any queue member or users above them in the role hierarchy can take ownership of records in a queue. Use [queues](#) to route custom object records to a group.

[Sharing Rules Overview](#)

With sharing rules, you can make automatic exceptions to your organization-wide sharing settings for defined sets of users. For example, use sharing rules to extend sharing access to users in public groups or roles. Sharing rules can never be stricter than your organization-wide default settings. They simply allow greater access for particular users.

You can create the following sharing rule.

Type	Based on	Set Default Sharing Access for
Custom object sharing rules	Custom object owner or other criteria, including custom object record types or field values	Individual custom object records

Sharing Rule Categories

When you define a sharing rule, you can choose from the following categories in the owned by members of and Share with drop-down lists. Depending on the type of sharing rule and the features enabled for your organization, some categories may not appear.

Category	Description
Queues	All records owned by the queue, excluding records owned by individual members of the queue. Available only in the owned by members of list.
Public Groups	All public groups defined by your administrator.
Roles	All roles defined for your organization. This includes all of the users in the specified role.

Sharing Rule Considerations

Sharing rules allow you to selectively grant data access to defined sets of users. Review the following notes before using sharing rules:

- You can use sharing rules to grant wider access to data. You cannot restrict access below your organization-wide default levels.
- Sharing rules apply to all new and existing records that meet the definition of the source data set.
- Sharing rules apply to both active and inactive users.
- When you change the access levels for a sharing rule, all existing records are automatically updated to reflect the new access levels.
- When you delete a sharing rule, the sharing access created by that rule is automatically removed.
- When you transfer records from one user to another, the sharing rules are reevaluated to add or remove access to the transferred records as necessary.
- When you modify which users are in a group or role, the sharing rules are reevaluated to add or remove access as necessary.
- Sharing rules automatically grant additional access to related records. For example, album sharing rules give role or group members access to the tracks associated with the shared album if they do not already have it.
- If multiple sharing rules give a user different levels of access to a record, the user gets the most permissive access level.
- Users in the role hierarchy are automatically granted the same access that users below them in the hierarchy have from a sharing rule, provided that the **Grant Access Using Hierarchies** option is selected.
- Making changes to sharing rules may require changing a large number of records at once.

Sharing Default Access Settings

User Permissions Needed	
To set default sharing access:	“Manage Users”

You can use organization-wide defaults to set the default level of record access for custom objects, with the following access levels.

Field	Description
Controlled by Parent	A user can perform an action (such as view, edit, or delete) on a child object based on whether he or she can perform that same action on the parent object associated with it in a master-detail relationship.
Private	Only the record owner, and users above that role in the hierarchy, can view and edit those records.
Public Read Only	All users can view records but not edit them. Only the owner, and users above that role in the hierarchy, can edit those records.
Public Read/Write	All users can view and edit all records.

Managing Criteria-Based Sharing Rules

Creating Custom Object Sharing Rules

User Permissions Needed	
To create sharing rules:	“Manage Users”

Custom object sharing rules can be based on the record owner or on other criteria, including certain field values. You can define up to 300 custom object sharing rules, including up to 50 criteria-based sharing rules.

1. If you plan to include public groups in your sharing rule, confirm that the appropriate groups have been created.
2. Click **Security Controls > Sharing Settings**.
3. In the Sharing Rules related list for the custom object, click **New**.
4. Enter the Label and Rule Name. The Label is the sharing rule label as it appears on the Database.com user interface. The Rule Name is a unique name used by the API.
5. Select a rule type.
6. Depending on the rule type you selected, do the following:
 - **Based on record owner**—In the **owned by members of** line, specify the users whose records will be shared: select a **category** from the first drop-down list and a set of users from the second drop-down list (or lookup field, if your organization has over 200 queues, groups, or roles).
 - **Based on criteria**—Specify the Field, Operator, and Value criteria that records must match to be included in the sharing rule. The fields available depend on the object selected, and the value is always a literal number or string. Click **Add Filter Logic...** to change the default AND relationship between each filter.



Note: To use a field that's not supported by criteria-based sharing rules, you can create a workflow rule or Apex trigger to copy the value of the field into a text or numeric field, and use that field as the criterion.

7. In the Share with line, specify the users who should have access to the data: select a category from the first drop-down list and a set of users from the second drop-down list or lookup field.
8. Select the sharing access setting for users.

Access Setting	Description
Read Only	Users can view, but not update, records.
Read/Write	Users can view and update records.

9. Click **Save**.

Editing Custom Object Sharing Rules

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

For sharing rules that are based on owner, you can edit only the sharing access settings. For sharing rules that are based on other criteria, you can edit the criteria and sharing access settings.

1. Click **Security Controls > Sharing Settings**.
2. In the Sharing Rules related list for the custom object, click **Edit** next to the rule you want to change.
3. Change the Label and Rule Name if desired.
4. If you selected a rule that's based on owner, skip to the next step.

If you selected a rule that's based on criteria, specify the criteria that records must match to be included in the sharing rule. The fields available depend on the object selected, and the value must be a literal number or string. Click **Add Filter Logic...** to change the default AND relationship between each filter.

5. Select the sharing access setting for users.

Access Setting	Description
Read Only	Users can view, but not update, records.
Read/Write	Users can view and update records.

6. Click **Save**.

Managing Sharing Calculations

Defer Sharing Calculations Overview

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

Performing a large number of configuration changes can lead to very long sharing rule evaluations or timeouts. To avoid these issues, an administrator can suspend these calculations and resume calculations during an organization's maintenance period.

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

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, see [Managing Group Membership Calculations](#).

To suspend, resume, or recalculate sharing rule calculation, see [Sharing Rule Recalculation](#).

Recalculating Sharing Rules

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

When you make changes to groups and roles, sharing rules are usually automatically reevaluated to add or remove access as necessary. Changes could include adding or removing individual users from a group or role, changing which role a particular role reports to, or adding or removing a group from within another group. However, if these changes affect too many records at once, a message appears warning that the sharing rules won't be automatically reevaluated, and you must manually recalculate them.

To manually recalculate an object's sharing rules:

- Click **Security Controls > Sharing Settings**.
- In the Sharing Rules related list for the object you want, click **Recalculate**.



Note: The **Recalculate** button is disabled when group membership or sharing rule calculations are deferred.

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

Automatic sharing rule calculation is enabled by default. You can [defer sharing rule calculation](#) by suspending and resuming at your discretion.

Sharing Rule Recalculation

User Permissions Needed

To suspend, resume or recalculate sharing rules: “Manage Users”

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

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.

Managing Group Membership Calculations

User Permissions Needed

To defer sharing calculations: “Manage Users”

AND

“View Setup and Configuration”

AND

“Manage Sharing Calculation Deferral”

To suspend and resume group membership calculation: “Manage Users”

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

When you make changes to roles, groups, or users, group membership is automatically recalculated to add or remove access as necessary. Changes can include adding or removing a user from a group or changing a role to allow access to different sets of reports.

If you are making changes to groups that affect a lot of records, you may want to suspend automatic group membership 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**.

Managing User Security

User Security Overview

Database.com provides each user in your organization with a unique username and password that must be entered each time a user logs in. Database.com issues a session cookie only to record encrypted authentication information for the duration of a specific session. The session cookie does not include either the username or password of the user. Database.com does not use cookies to store other confidential user and session information, but instead implements more advanced security methods based on dynamic data and encoded session IDs.

User Authentication

Database.com has its own system of user authentication, but some companies prefer to use an existing single sign-on capability to simplify and standardize their user authentication. You have two options to implement single sign-on—federated authentication using Security Assertion Markup Language (SAML) or delegated authentication.

- Federated authentication using Security Assertion Markup Language (SAML) allows you to send authentication and authorization data between affiliated but unrelated Web services. This enables you to sign-on to Database.com from a client application. Federated authentication using SAML is enabled by default for your organization.
- Delegated authentication single sign-on enables you to integrate Database.com with an authentication method that you choose. This enables you to integrate authentication with your LDAP (Lightweight Directory Access Protocol) server, or perform single sign-on by authenticating using a token instead of a password. You manage delegated authentication at the permission level, allowing some users to use delegated authentication, while other users continue to use their Database.com-managed password. Delegated authentication is set by permissions, not by organization. You must request that this feature be enabled by salesforce.com. Contact salesforce.com to enable delegated authentication single sign-on for your organization.

The primary reasons for using delegated authentication include:

- ◊ Using a stronger type of user authentication, such as integration with a secure identity provider
- ◊ Making your login page private and not part of the general Internet, but rather, part of your corporate network, behind your corporate firewall
- ◊ Differentiating your organization from all other applications that use Database.com in order to reduce phishing attacks

For more information, see [About Single Sign-On](#) on page 289.

Managing Login Restrictions

Setting Login Restrictions

To help protect your organization's data against unauthorized access, you have several options for setting login restrictions.

Login Hours

For each profile, you can set the hours when users can log in. See:

- [Viewing and Editing Login Hours in the Enhanced Profile User Interface](#) on page 259
- [Viewing and Editing Login Hours in the Original Profile User Interface](#) on page 258

Login IP Address Ranges

For each profile, you can set the IP addresses from which users can log in. See:

- [Viewing and Editing Login IP Ranges in the Enhanced Profile User Interface](#) on page 258
- [Viewing and Editing Login IP Address Ranges in the Original Profile User Interface](#) on page 257

Organization-Wide Trusted IP Address List

For all users, you can set a list of IP address ranges from which they can always log in without receiving a login challenge.

See [Restricting Login IP Ranges for Your Organization](#) on page 241.

When users log in to Database.com, either via the user interface, the API, or the Data Loader, Database.com confirms that the login is authorized as follows:

1. Database.com checks whether the user's profile has login hour restrictions. If login hour restrictions are specified for the user's profile, any login outside the specified hours is denied.
2. Database.com then checks whether the user's profile has IP address restrictions. If IP address restrictions are defined for the user's profile, any login from an undesignated IP address is denied, and any login from a specified IP address is allowed.
3. If profile-based IP address restrictions are not set, Database.com checks whether the user is logging in from an IP address they have not used to access Database.com before:
 - If the user's login is from a browser that includes a Database.com cookie, the login is allowed. The browser will have the Database.com cookie if the user has previously used that browser to log in to Database.com, and has not cleared the browser cookies.
 - If the user's login is from an IP address in your organization's trusted IP address list, the login is allowed.
 - If the user's login is from neither a trusted IP address nor a browser with a Database.com cookie, the login is blocked.

Whenever a login is blocked or returns an API login fault, Database.com must verify the user's identity:

- For access via the user interface, the user is prompted to click a **Send Activation Link** button to send an activation email to the address specified on the user's Database.com record. The email instructs the user to copy and paste an activation link into their browser to activate their computer for logging in to Database.com. The activation link included in the email is valid for up to 24 hours from the time the user clicked the **Send Activation Link** button. After 24 hours, the activation link expires, and users must repeat the activation process to log in.



Note: A user will not be asked to activate the first time they log in to Database.com.

- For access via the API or a client, the user must add their security token to the end of their password in order to log in. A security token is an automatically-generated key from Database.com. For example, if a user's password is mypassword, and their security token is XXXXXXXXXXXX, then the user must enter mypasswordXXXXXXXXXX to log in.

Users can obtain their security token by changing their password or resetting their security token via the Database.com user interface. When a user changes their password or resets their security token, Database.com sends a new security token to the email address on the user's Database.com record. The security token is valid until a user resets their security token, changes their password, or has their password reset.



Tip: We recommend that you obtain your security token using the Database.com user interface from a trusted network prior to attempting to access Database.com from a new IP address.

Tips on Setting Login Restrictions

Consider the following when setting login restrictions:

- When a user's password is changed, the user's security token is automatically reset. The user may experience a blocked login until he or she adds the automatically-generated security token to the end of his or her password when logging in to Database.com via the API or a client.
- For more information on API login faults, see the Core Data Types Used in API Calls topic in the [Web Services API Developer's Guide](#).
- If single sign-on is enabled for your organization, API users can't log into Database.com unless their IP address is included on your organization's list of trusted IP addresses or on their profile, if their profile has IP address restrictions set. Furthermore, the single sign-on authority usually handles login lockout policies for users with the "Is Single Sign-On Enabled" permission. However, if the security token is enabled for your organization, then your organization's login lockout settings determine the number of times a user can attempt to log in with an invalid security token before being locked out of Database.com.
- The following events count toward the number of times a user can attempt to log in with an invalid password before being locked out of Database.com, as defined in your organization's login lockout settings:
 - ◊ Each time a user is prompted to click the **Send Activation Link** button
 - ◊ Each time a user incorrectly adds their security token to the end of their password to log into the API or a client

Network-Based Security

User authentication determines who can log in, while *network-based security* limits where they can log in from and when. Use network-based security to limit the window of opportunity for an attacker by restricting the origin of user logins. Network-based security can also make it more difficult for an attacker to use stolen credentials.

To enhance network-based security, Database.com includes the ability to restrict the hours during which users can log in and the range of IP addresses from which they can log in. If IP address restrictions are defined for a user's profile and a login originates from an unknown IP address, Database.com does not allow the login. This helps to protect your data from unauthorized access and "phishing" attacks.

To set the organization-wide list of trusted IP addresses from which users can always log in without a login challenge, see [Restricting Login IP Ranges for Your Organization](#) on page 241. To restrict login hours by profile, or to restrict logins by IP addresses for specific profiles, see [Setting Login Restrictions](#) on page 255.

Viewing and Editing IP Address Ranges

Viewing and Editing Login IP Address Ranges in the Original Profile User Interface

User Permissions Needed	
To set login IP ranges:	"Manage Users"

You can set the IP addresses from which users with a particular profile can log in. When you define IP address restrictions for a profile, any login from an undesignated IP address is denied, and any login from a specified IP address is allowed. To set IP addresses on profiles:

1. Click **Manage Users** > **Profiles** and select a profile.
2. Click **New** in the Login IP Ranges related list.

3. Enter a valid IP address in the IP Start Address and a higher IP address in the IP End Address field.

The start and end addresses define the range of allowable IP addresses from which users can log in. To allow logins from a single IP address, enter the same address in both fields. For example, to allow logins from only 125.12.3.0, enter 125.12.3.0 as both the start and end addresses.

4. Click Save.

Both IP addresses in a range must be either IPv4 or IPv6. In ranges, IPv4 addresses exist in the IPv4-mapped IPv6 address space ::ffff:0:0 to ::ffff:ffff:ffff, where ::ffff:0:0 is 0.0.0.0 and ::ffff:ffff:ffff is 255.255.255.255. A range can't include IP addresses inside of the IPv4-mapped IPv6 address space if it also includes IP addresses outside of the IPv4-mapped IPv6 address space. Ranges such as 255.255.255.255 to ::1:0:0:0 or :: to ::1:0:0:0 are not allowed. You can set up IPv6 addresses in all organizations, but IPv6 is only enabled for login in test database organizations for the Spring '12 release.

Viewing and Editing Login IP Ranges in the Enhanced Profile User Interface

User Permissions Needed	
To view login IP ranges:	"View Setup and Configuration"
To edit login IP ranges:	"Manage Users"

For each profile, you can view and specify the IP addresses from which users can log in. When you define IP address restrictions for a profile, any login from an undesignated IP address is denied, and any login from a specified IP address is allowed. To view and edit IP address ranges in the enhanced profile user interface:

1. Click Manage Users > Profiles.

2. Select a profile.

3. In the profile overview page, click Login IP Ranges.

4. Use any of these methods to change login IP address ranges for the profile.

- **Click Add IP Ranges.** Enter a valid IP address in the IP Start Address and a higher IP address in the IP End Address field. The start and end addresses define the range of allowable IP addresses from which users can log in. To allow logins from a single IP address, enter the same address in both fields. For example, to allow logins from only 125.12.3.0, enter 125.12.3.0 as both the start and end addresses.
- To change or remove an existing IP address range, click **Edit** or **Delete** for that range.

Both IP addresses in a range must be either IPv4 or IPv6. In ranges, IPv4 addresses exist in the IPv4-mapped IPv6 address space ::ffff:0:0 to ::ffff:ffff:ffff, where ::ffff:0:0 is 0.0.0.0 and ::ffff:ffff:ffff is 255.255.255.255. A range can't include IP addresses inside of the IPv4-mapped IPv6 address space if it also includes IP addresses outside of the IPv4-mapped IPv6 address space. Ranges such as 255.255.255.255 to ::1:0:0:0 or :: to ::1:0:0:0 are not allowed. You can set up IPv6 addresses in all organizations, but IPv6 is only enabled for login in test database organizations for the Spring '12 release.

Viewing and Editing Login Hours

Viewing and Editing Login Hours in the Original Profile User Interface

User Permissions Needed	
To set login hours:	“Manage Users”

For each profile, you can specify the hours when users can log in.

1. Click **Manage Users > Profiles**, and select a profile.
2. Click **Edit** in the Login Hours related list.
3. Set the days and hours when users with this profile can use the system.

To allow users to log in at any time, click **Clear All Times**. To prohibit users from using the system on a specific day, set the start and end times to the same value.

If a user logs in before the restricted hours, the system ends the user's session when the restricted hours begin.

4. Click **Save**.



Note: The first time login hours are set for a profile, the hours are based on the organization's Default Time Zone as specified at **Company Profile > Company Information**. After that, any changes to the organization's Default Time Zone won't change the time zone for the profile's login hours. As a result, the login hours are always applied at those exact times even if a user is in a different time zone or if the organization's default time zone is changed.

Depending on whether you're viewing or editing login hours, the hours may appear differently. On the profile detail page, hours are shown in your specified time zone. On the Login Hours edit page, they appear in the organization's default time zone.

Viewing and Editing Login Hours in the Enhanced Profile User Interface

User Permissions Needed	
To view login hour settings:	“View Setup and Configuration”
To edit login hour settings:	“Manage Users”

For each profile, you can specify the hours when users can log in.

1. Click **Manage Users > Profiles**.
2. Select a profile.
3. In the profile overview page, click **Login Hours**.
4. To change the login hours, click **Edit**.
5. Set the days and hours when users with this profile can log in to the organization.

To allow users to log in at any time, click **Clear All Times**. To prohibit users from using the system on a specific day, set the start and end times to the same value.

If a user logs in before the restricted hours, the system ends the user's session when the restricted hours begin.



Note: The first time login hours are set for a profile, the hours are based on the organization's Default Time Zone as specified at **Company Profile > Company Information**. After that, any changes to the organization's Default Time Zone won't change the time zone for the profile's login hours. As a result, the login hours are always applied at those exact times even if a user is in a different time zone or if the organization's default time zone is changed.

Depending on whether you're viewing or editing login hours, the hours may appear differently. On the Login Hours detail page, hours are shown in your specified time zone. On the Login Hours Edit page, they appear in the organization's original default time zone.

Managing Groups

About Groups

Groups, also known as public groups, are sets of users. They can contain individual users, other groups, the users in a particular role, or the users in a particular role plus all of the users below that role in the hierarchy.

Only administrators can create public groups, but they can be used by everyone in the organization.

You can use public groups by setting up default sharing access via a [sharing rule](#).

Group Member Types

User Permissions Needed	
To create or edit a public group:	"Manage Users"

When you create or edit a group, you can select the following types of members from the [Search](#) drop-down list. Depending on your organization settings, some types may not be available.

Member Type	Description
Public Groups	All public groups defined by your administrator.
Roles	All roles defined for your organization. Adding a role to a group includes all of the users in that role.
Roles and Subordinates	Adding a role and its subordinate roles includes all of the users in that role plus all of the users in roles below that role.
Users	All users in your organization.

Creating and Editing Groups

User Permissions Needed	
To create or edit a public group:	"Manage Users"

Only administrators can create and edit public groups.

To create or edit a group:

1. Click **Manage Users > Public Groups**.
2. Click **New**, or click **Edit** next to the group you want to edit.
3. Enter the following:

Field	Description

Label	The name used to refer to the group in any user interface pages.
API Name	The unique name used by the API. The name must begin with a letter and use only alphanumeric characters and underscores.
Grant Access Using Hierarchies	Select Grant Access Using Hierarchies to allow automatic access to records using your role or hierarchies. When selected, any records shared with users in this group are also shared with users higher in the hierarchy.
	 Note: If Grant Access Using Hierarchies is deselected, users that are higher in the role hierarchy don't receive automatic access. However, some users—such as those with the “View All” and “Modify All” object permissions and the “View All Data” and “Modify All Data” system permissions—can still access records they don't own.
Search	From the Search drop-down list, select the type of member to add. If you don't see the member you want to add, enter keywords in the search box and click Find .
Selected Members	Select members from the Available Members box, and click Add to add them to the group.

4. Click **Save**.



Note: When you edit groups and roles, sharing rules are usually automatically reevaluated to add or remove access as needed. If these changes affect too many records at once, a message appears warning that the sharing rules won't be automatically reevaluated, and you must manually recalculate them.

Viewing All Users in a Group

The All Users list shows users who belong to the selected [public group](#), [queue](#), or [role sharing group](#). From this page, you can view detailed user information, edit user information, and access related information.

- To show a filtered list of items, select a predefined list from the View drop-down list, or click **Create New View** to define your own [custom view](#). To edit or delete any view you created, select it from the View drop-down list and click **Edit**.
- Click **Edit** next to a username to [edit the user information](#).
- Click **Login** next to a username to [log in as that user](#). This link is only available if the user has granted you login access.

Viewing Group Lists

User Permissions Needed	
To edit a public group:	“Manage Users”

To view [public group](#) lists:

- Click **Manage Users > Public Groups**.
- To show a filtered list of items, select a predefined list from the View drop-down list, or click **Create New View** to define your own [custom view](#). To edit or delete any view you created, select it from the View drop-down list and click **Edit**.

- Click the name of a group in the Groups related list to display the group's detail page.

- To [edit the group membership](#), click **Edit**.
- To delete the group, click **Delete**.
- To view group members, see the Group Members related list.
- To view group members and users who have equivalent access because they are higher in the [role](#), click **View All Users** to display the All Users in Group related list. Click **View Group Members** to return to the Group Members related list.

Managing Record Security

Granting Access to Records

Users can manually grant access to certain types of records. In some cases, granting access includes access to all associated records. For example, if you grant another user access to your Job Position object, the user will automatically have access to all the child objects such as Candidate and Hiring Manager associated with that account. To grant sharing privileges for a record, you must be the record owner, any user granted Full Access, an administrator, or (provided your sharing settings control access through hierarchies) a user in a role above the owner in the hierarchy.

To grant access to a record:

- Click **Sharing** on the record you want to share.
- Click **Add**.
- From the drop-down list, select the type of group, user, or role to add.

Your choices are:

Type	Description
Public Groups	All public groups defined by your administrator.
Personal Groups	All personal groups defined by the record owner. Only the record owner can share with his or her personal groups.
Users	All users in your organization.
Roles	All roles defined for your organization. This includes all of the users in that role.
Roles and Subordinates	This includes all of the users in the role plus all of the users in roles below that role.

- Choose the specific groups, users, roles, or territories who should have access by adding their names to the New Sharing list.
- Choose the access level for the record you are sharing and any associated records that you own.

The possible access levels are:

Access Level	Description
Full Access	User can view, edit, delete, and transfer the record. User can also extend sharing access to other users; however, the user cannot grant Full Access to other users.

Access Level	Description
Read/Write	User can view and edit the record, and add associated records, notes, and attachments to it.
Read Only	User can view the record, and add associated records to it. They cannot edit the record or add notes or attachments.
Private	User cannot access the record in any way.

**Note:**

- Contact Access is not available when the organization-wide default for contacts is set to Controlled by Parent.
- For sharing rules that specify access for associated object records, the given access level applies to that sharing rule only. For example, if an account sharing rule specifies Private as the access level for associated contacts, a user may still have access to associated contacts via other means, such as [organization-wide defaults](#), the “Modify All Data” or “View All Data” [permission](#), or the “Modify All” or “View All” [permission](#) for contacts.

6. Select the reason for the share to allow users and administrators to understand the source of the sharing. For more information on sharing reasons, see “Creating Apex Sharing Reasons” in the online help.
7. Click **Save**.

Editing or Deleting Record Access

To edit the access levels for a record, click **Sharing** on the record, and then click **Edit** next to the group, user, or role whose access you want to modify. To delete the sharing access for a group, user, or role, click **Sharing** on the record, and then click **Del** next to the group, user, or role whose access you want to remove.

Managing User Access to Fields**Managing Field Accessibility****About Field Accessibility**

Several factors help control whether users can view and edit specific fields in Database.com.

- **Field-level security**—You can further restrict users' access to fields by setting whether those fields are visible, editable, or read only.
- **Permissions**—Some user permissions override field-level security settings. For example, users with the “Edit Read Only Fields” permission can always edit read-only fields regardless of any other settings.
- **Universally required fields**—A custom field can be made [universally required](#), which overrides any less-restrictive settings on field-level security.

After setting these items, you can confirm users' access to specific fields using the [field accessibility grid](#).

Checking Field Accessibility for a Particular Field

User Permissions Needed	
To view field accessibility:	“View Setup and Configuration”

1. Navigate to the appropriate object:

- Click **Create > Objects**, and click the name of the custom object in the list.
2. Select a field and click **View Field Accessibility**.
 3. Confirm that the field access is correct for different profiles.
 4. Hover your mouse over any field access setting to see whether the field is required, editable, hidden, or read only based on field-level security.
 5. Click any field access setting to [change the field's accessibility](#).

For advanced options to check field accessibility by a specific profile, record type, or field, click **Security Controls > Field Accessibility**. From this page, you need to choose a type of record to view and then select whether you want to check access by profiles or fields.

 **Note:** In this user interface, you can't check access for permission sets.

Modifying Field Access Settings

User Permissions Needed	
To view field accessibility:	“View Setup and Configuration”
To change field accessibility:	“Customize Application” AND “Manage Users”

From the [field accessibility grid](#), you can click any field access setting to change the field's accessibility in field-level security. The Access Settings page then lets you modify the field access settings.

- In the Field-Level Security section of the page, specify the field's access level for the profile.

Access Level	Enabled Settings
Users can read and edit the field.	Visible
Users can read but not edit the field.	Visible and Read-Only
Users can't read or edit the field.	None

Managing Field-Level Security

Field-Level Security Overview

Field-level security settings let administrators restrict users' access to view and edit specific fields in:

- Detail and edit pages
- Related lists
- List views
- Synchronized data
- Imported data



Important: To set up your organization to prevent users from retrieving records that match a value in a field hidden by field-level security, contact salesforce.com Customer Support.

You can define field-level security [for multiple fields on a single permission set or profile](#)

Setting Field Permissions in Permission Sets and Profiles

User Permissions Needed	
To set field-level security:	“Customize Application”

1. Click **Manage Users**, then click **Permission Sets or Profiles**.
2. Select a permission set or profile.
3. Depending on which interface you're using, do one of the following:
 - Permission sets or enhanced profile user interface—In the **Find Settings...** box, enter the name of the object you want and select it from the list. Click **Edit**, then scroll to the Field Permissions section.
 - Original profile user interface—In the Field-Level Security section, click **View** next to the object you want to modify, and then click **Edit**.
4. [Specify the field's access level](#).
5. Click **Save**.

Field Permissions

Field permissions specify the access level for each field in an object. In permission sets and the enhanced profile user interface, the setting labels differ from those in the original profile user interface and in field-level security pages for customizing fields.

Access Level	Enabled Settings in Permission Sets and Enhanced Profile User Interface	Enabled Settings in Original Profile and Field-Level Security Interfaces
Users can read and edit the field.	Read and Edit	Visible
Users can read but not edit the field.	Read	Visible and Read-Only
Users can't read or edit the field.	None	None

Managing Connection Security

[Managing SAML](#)

[About SAML](#)

User Permissions Needed	
To view the settings:	“View Setup and Configuration”
To edit the settings:	“Customize Application” AND “Modify All Data”

Security Assertion Markup Language (SAML) is an XML-based standard that allows you to communicate authentication decisions between one service and another. It underlies many Web single sign-on solutions. Database.com supports SAML for single sign-on into Database.com from a corporate portal or identity provider.

Much of the work to set up single sign-on using SAML occurs with your identity provider:

1. Establish a SAML identity provider and [gather information](#) about how they will connect to Database.com. This is the provider that will send single sign-on requests to Database.com.
2. Provide information to your identity provider, such as the [URLs for the start and logout pages](#).
3. Configure Database.com using the instructions in [Configuring SAML Settings for Single Sign-On](#). This is the only step that takes place in Database.com.

Your identity provider should send SAML assertions to Database.com using the SAML Web Single Sign-on Browser POST profile. Database.com sends SAML responses to the [Identity Provider Login URL](#) specified in **Security Controls > Single Sign-On Settings**. Database.com receives the assertion, verifies it against your Database.com configuration, and allows single sign-on if the assertion is true.

If you have problems with the SAML assertion after you configure Database.com for SAML, use the SAML Assertion Validator to [validate the SAML assertion](#). You may need to obtain a SAML assertion from your identity provider.

If your users are having problems using SAML to login, you can [review the SAML login history](#) to determine why they were not able to log in and share that information with your identity provider.

If your identity provider supports metadata, and if you've configured SAML using version 2.0, you can click **Download Metadata** to download an XML configuration file to send them, which they can then upload to automatically configure their settings for connecting to your Database.com organization.

Working With Your Identity Provider

User Permissions Needed	
To view the settings:	“View Setup and Configuration”
To edit the settings:	“Customize Application” AND “Modify All Data”

1. You must gather the following information from your identity provider before configuring Database.com for SAML.
 - The version of SAML the identity provider uses (1.1 or 2.0)
 - The entity ID of the identity provider (also known as the issuer)
 - An authentication certificate.



Tip: Be sure to store the certificate where you can access it from your browser. This will be uploaded to Database.com in a later step.

- The following SAML assertion parameters, as appropriate:
 - ◊ The SAML user ID type
 - ◊ The SAML user ID location
 - ◊ Attribute Name
 - ◊ Attribute URI
 - ◊ Name ID format



Note: Attribute Name, Attribute URI, and Name ID format are only necessary if the [SAML User ID Location](#) is in an Attribute element, and not the name identifier element of a Subject statement.

You may also want to share [more information](#) about these values with your identity provider.

 **Tip:** Enable Database.com for SAML and take a screenshot of the page for your identity provider. Click **Security Controls > Single Sign-On Settings**, click **Edit**, then select **SAML Enabled**.

2. Work with your identity provider to setup the [start, login, and logout pages](#).
3. Share the [example SAML assertions](#) with your identity provider so they can determine the format Database.com requires for successful single sign-on.

Identity Provider Values

User Permissions Needed	
To view the settings:	“View Setup and Configuration”
To edit the settings:	“Customize Application” AND “Modify All Data”

Before you can configure Database.com for SAML, you must receive information from your identity provider. This information must be used on the [single sign-on page](#).

The following information might be useful for your identity provider.

Field	Description
SAML Version	The version of SAML your identity provider uses. Database.com currently supports version 1.1 and 2.0. The SAML specifications for the various versions are linked below: <ul style="list-style-type: none"> • SAML 1.1 • SAML 2.0

Field	Description
Issuer	The Entity ID—a URL that uniquely identifies your SAML identity provider. SAML assertions sent to Database.com must match this value exactly in the <saml:Issuer> attribute of SAML assertions.
Entity ID	The issuer in SAML requests generated by Database.com, and is also the expected audience of any inbound SAML Responses. If you don't have domains deployed, this value can be https://saml.salesforce.com or https://saml.database.com . If you have domains deployed, Database.com recommends that you use your custom domain name. You can find the value on the Single Sign-On Settings page. Click Security Controls > Single Sign-On Settings .
Identity Provider Certificate	The authentication certificate issued by your identity provider.
Identity Provider Login URL	For SAML 2.0 only: The URL where Database.com sends a SAML request to start the login sequence. If you have domains deployed and a value specified for this field, login requests are usually sent to the address specified by this field. However, if you need to bypass this value (for example, your identity provider is down) add the <code>login</code> parameter to the query string for the login page. For example: http://mydomain.database.com?login .
Identity Provider Logout URL	For SAML 2.0 only: The URL to direct the user to when they click the Logout link in Database.com. The default is http://www.salesforce.com .
Database.com Login URL	The URL associated with login for the Web single sign-on flow. See SAML Assertion Flow on page 345.
OAuth 2.0 Token Endpoint	For SAML 2.0 only: The ACS URL used with enabling Database.com as an identity provider in the Web single sign-on OAuth assertion flow. See SAML Assertion Flow on page 345.
Custom Error URL	The URL of the page users should be directed to if there's an error during SAML login. It must be a publicly accessible page. The URL can be absolute or relative.
SAML User ID Type	The element in a SAML assertion that contains the string that identifies a Database.com user. Values are:
Assertion contains User's Salesforce username	
Use this option if your identity provider passes the Database.com username in SAML assertions.	
Assertion contains the Federation ID from the User object	
Use this option if your identity provider passes an external user identifier, for example an employee ID, in the SAML assertion to identify the user.	
SAML User ID Location	The location in the assertion where a user should be identified. Values are:
User ID is in the NameIdentifier element of the Subject statement	
The Database.com Username or FederationIdentifier is located in the <Subject> statement of the assertion.	

Field	Description
	<p>User ID is in an Attribute element</p> <p>The Database.com Username or FederationIdentifier is specified in an <AttributeValue>, located in the <Attribute> of the assertion.</p>
Attribute Name	If “User ID is in an Attribute element” is selected, this contains the value of the AttributeName that is specified in <Attribute> that contains the User ID.
Attribute URI	If SAML 1.1 is the specified SAML version and “User ID is in an Attribute element” is selected, this contains the value of the AttributeNamespace that is specified in <Attribute>.
Name ID Format	If SAML 2.0 is the specified SAML version and “User ID is in an Attribute element” is selected, this contains the value for the nameid-format. Possible values include unspecified, emailAddress or persistent. All legal values can be found in the “Name Identifier Format Identifiers” section of the Assertions and Protocols SAML 2.0 specification .

Start, Login and Logout URL Values

In addition to the information used during the single sign-on, your identity provider can also set the start, login, and logout pages. You can also specify these pages yourself when you [configure single sign-on](#).

The following information might be useful to your identity provider if they are setting these pages.

- The SAML specification supports an HTML form that is used to pass the SAML assertion via HTTPS POST.
- For SAML 1.1, the SAML identity provider can embed name-value pairs in the TARGET field to pass this additional information to Database.com prepended with a specially formatted URL that contains URL-encoded parameters.
- The URL for SAML 1.1 to include in the TARGET field can be either `https://saml.salesforce.com/?` or `https://saml.database.com/?`
- For SAML 2.0, instead of using the TARGET field, the identity providers uses the <AttributeStatement> in the SAML assertion to specify the additional information.
- Database.com supports the following parameters:



Note: For SAML 1.1 these parameters must be URL-encoded. This allows the URLs, passed as values that include their own parameters, to be handled correctly. For SAML 2.0, these parameters are part of the <AttributeStatement>.

- ◊ `ssoStartPage` is the page to which the user should be redirected when trying to log in with SAML. The user is directed to this page when requesting a protected resource in Database.com, without an active session. The `ssoStartPage` should be the SAML identity provider's login page.
- ◊ `startURL` is the URL where you want the user to be directed when sign-on completes successfully. This URL can be absolute, such as `https://na1.database.com/001/o` or it can be relative, such as `/001/o`. This parameter is only used in SAML 1.1. In SAML 2.0, the start URL is the page the user attempted to access before they were authenticated.
- ◊ `logoutURL` is the URL where you want the user to be directed when they click the **Logout** link in Database.com. The default is `http://www.salesforce.com`.

The following sample TARGET field is for SAML 1.1, and includes properly-encoded parameters. It passes a customized start page, as well as start and logout URLs embedded as parameter values in the query string.

```
https://saml.database.com/?ssoStartPage=https%3A%2F%2Fwww.customer.org%2Flogin%2F&startURL=%2F001%2Fo&logoutURL=http%3A%2F%2Fwww.salesforce.com
```

The following is an example of an <AttributeStatement> for SAML 2.0 that contains both ssoStartPage and logoutURL:

```

<saml:AttributeStatement>
    <saml:Attribute Name="ssoStartPage"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
        <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:anyType">
            http://www.customer.org
        </saml:AttributeValue>
    </saml:Attribute>

    <saml:Attribute Name="logoutURL"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
        <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string">
            https://www.salesforce.com
        </saml:AttributeValue>
    </saml:Attribute>
</saml:AttributeStatement>

```

Customizing SAML Start, Error, Login, and Logout Pages

User Permissions Needed	
To view the settings:	“View Setup and Configuration”
To edit the settings:	“Customize Application” AND “Modify All Data”

The start, error, login, and logout pages can be customized for single sign-on users using SAML 1.1 or 2.0. As part of your configuration, you need to decide the following:

- The logout page: the URL to direct the user to when they click the **Logout** link in Database.com. The default is <http://www.salesforce.com>.
- If your identity provider uses SAML 1.1, the start page: the URL to direct the user to when sign-on successfully completes. This URL can be absolute, such as <https://na1.database.com/001/o> or it can be relative, such as [/001/o](#). This URL should be an endpoint that accepts SAML authentication requests.

In SAML 2.0, the start page is the page the user attempted to access before they were authenticated. The SAML 2.0 start page must support Sp-init single sign-on.

If you are using SAML 2.0, you can also use the `RelayState` parameter to control where users get redirected after a successful login.

- The single sign-on start page where Database.com sends a SAML request to start the login sequence.

We recommend that if you specify a single sign-on start page that you also specify a logout page. When you specify a logout page, when a user clicks logout or if a user's session expires, the user is redirected to that page. If you don't specify a logout page, the user is redirected to the general Database.com login page.

For SAML 2.0, these values can be set either during the single sign-on configuration, or by your identity provider in the login URL or SAML assertion. The order of precedence is:

1. Session cookie—if you've already logged into Database.com and a cookie still exists, the login and logout pages specified by the session cookie are used.
2. Values passed in from the identity provider.
3. Values from the single sign-on configuration page.

If you decide not to add these values to the single sign-on configuration, share them with your identity provider. They will need to [use these values](#) either in the login URL or the assertion.

You can also decide if you want users to be directed to a custom error page if there's an error during SAML login: It must be a publicly accessible page. The URL can be absolute or relative. Use this value when you [configure SAML](#).

Example SAML Assertions

User Permissions Needed	
To view the settings:	“View Setup and Configuration”
To edit the settings:	“Customize Application” AND “Modify All Data”

Share the example SAML assertions with your identity provider so they can determine the format of the information Database.com requires for successful single-sign on.

In addition to the general single sign-on examples for both SAML 1.1 and SAML 2.0, use the following samples for the specific feature:

- [SOAP message for delegated authentication](#)
- [assertion for just-in-time provisioning](#)

SAML User ID type is the Salesforce or Database.com username, and SAML User ID location is the <NameIdentifier> element in the <Subject> element

SAML 1.1:

```
<Subject>
    <NameIdentifier>user101@database.com</NameIdentifier>
</Subject>
```

SAML 2.0:

```
<saml:Subject>
    <saml:NameID
        Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">user101@database.com</saml:NameID>
        <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
            <saml:SubjectConfirmationData NotOnOrAfter="2008-06-26T02:44:24.173Z"
                Recipient="http://localhost:9000"/>
            </saml:SubjectConfirmation>
        </saml:Subject>
```

SAML User ID type is the Salesforce or Database.com username, and SAML User ID location is the <Attribute> element

SAML 1.1:

```
<AttributeStatement>
  <Subject>
    <NameIdentifier>this value doesn't matter</NameIdentifier>
    <SubjectConfirmation>
      <ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer</ConfirmationMethod>
    </SubjectConfirmation>
  </Subject>
  <Attribute AttributeName="MyDbName" AttributeNamespace="MyDbURI">
    <AttributeValue>user101@database.com</AttributeValue>
  </Attribute>
</AttributeStatement>
```

SAML 2.0:

```
<saml:AttributeStatement>
  <saml:Attribute FriendlyName="fooAttrib" Name="DB_USERNAME"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
    <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string">
      user101@database.com
    </saml:AttributeValue>
  </saml:Attribute>
</saml:AttributeStatement>
```

SAML User ID type is the Database.com User object's FederationIdentifier field, and SAML User ID location is the <NameIdentifier> element in the <Subject> element

SAML 1.1:

```
<AttributeStatement>
  <saml:Subject>
    <saml:NameIdentifier Format="urn:oasis:names:tc:SAML:1.0:assertion"
    NameQualifier="www.saml_assertions.com">
      MyName
    </saml:NameIdentifier>
  </saml:Subject>
</AttributeStatement>
```

SAML 2.0:

```
<saml:Subject>
  <saml:NameID
  Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">MyName</saml:NameID>
  <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
    <saml:SubjectConfirmationData NotOnOrAfter="2008-06-26T02:48:25.730Z"
    Recipient="http://localhost:9000/">
  </saml:SubjectConfirmation>
</saml:Subject>
```



Note: The name identifier can be any arbitrary string, including email addresses or numeric ID strings.

SAML User ID type is the Database.com User object's Federation Identifier field, and SAML User ID location is the <Attribute> element

SAML 1.1:

```
<AttributeStatement>
  <Subject>
    <NameIdentifier>who cares</NameIdentifier>
    <SubjectConfirmation>
      <ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer</ConfirmationMethod>
    </SubjectConfirmation>
  </Subject>
  <Attribute AttributeName="MyName" AttributeNamespace="MyURI">
    <AttributeValue>user101</AttributeValue>
  </Attribute>
</AttributeStatement>
```

SAML 2.0:

```
<saml:AttributeStatement>
  <saml:Attribute FriendlyName="fooAttrib" Name="DB_ATTR"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
    <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string">
      user101
    </saml:AttributeValue>
  </saml:Attribute>
</saml:AttributeStatement>
```

SAML User ID type is the Salesforce or Database.com username, and SAML User ID location is the <NameIdentifier> element in the <Subject> element

The following is a complete SAML response, for SAML 2.0:

```
<samlp:Response ID="_257f9d9e9fa14962c0803903a6ccad931245264310738"
  IssueInstant="2009-06-17T18:45:10.738Z" Version="2.0">
<saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
  https://www.database.com</saml:Issuer>

<samlp:Status>
  <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
</samlp:Status>

<saml:Assertion ID="_3c39bc0fe7b13769cab2f6f45eba801b1245264310738"
  IssueInstant="2009-06-17T18:45:10.738Z" Version="2.0">
  <saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
    https://www.database.com</saml:Issuer>

<saml:Subject>
  <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
    saml01@database.com</saml:NameID>

  <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
    <saml:SubjectConfirmationData NotOnOrAfter="2009-06-17T18:50:10.738Z"
      Recipient="https://login.database.com"/>
  </saml:SubjectConfirmation>
</saml:Subject>

<saml:Conditions NotBefore="2009-06-17T18:45:10.738Z"
  NotOnOrAfter="2009-06-17T18:50:10.738Z">
```

```

<saml:AudienceRestriction>
  <saml:Audience>https://saml.database.com</saml:Audience>
</saml:AudienceRestriction>
</saml:Conditions>

<saml:AuthnStatement AuthnInstant="2009-06-17T18:45:10.738Z">
  <saml:AuthnContext>
    <saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified
    </saml:AuthnContextClassRef>
  </saml:AuthnContext>
</saml:AuthnStatement>

<saml:AttributeStatement>

  <saml:Attribute Name="organization_id">
    <saml:AttributeValue xsi:type="xs:anyType">00DD0000000F7L5
    </saml:AttributeValue>
  </saml:Attribute>

  <saml:Attribute Name="ssostartpage"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
    <saml:AttributeValue xsi:type="xs:anyType">
      http://www.database.com/security/saml/saml20-gen.jsp
    </saml:AttributeValue>
  </saml:Attribute>

  <saml:Attribute Name="logouturl"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
    <saml:AttributeValue xsi:type="xs:string">
      http://www.database.com/security/del_auth/SsoLogoutPage.html
    </saml:AttributeValue>
  </saml:Attribute>
</saml:AttributeStatement>
</saml:Assertion>
</samlp:Response>

```

Sample SOAP Message for Delegated Authentication

As part of the delegated authentication single sign-on process, a salesforce.com server makes a SOAP 1.1 request to authenticate the user who is passing in the credentials. Here is an example of this type of request. Your single sign-on Web service needs to accept this request, process it, and return a true or false response.

Sample Request

```

<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <Authenticate xmlns="urn:authentication.soap.sforce.com">
      <username>sampleuser@sample.org</username>
      <password>myPassword99</password>
      <sourceIp>1.2.3.4</sourceIp>
    </Authenticate>
  </soapenv:Body>
</soapenv:Envelope>

```

Sample Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Body>
<AuthenticateResult
xmlns="urn:authentication.soap.sforce.com">
<Authenticated>false</Authenticated>
</AuthenticateResult>
</soapenv:Body>
</soapenv:Envelope>
```

Sample SAML Assertion for Just-In-Time Provisioning

The following is a sample SAML assertion for just in time provisioning.

```
<saml:Attribute Name="User.Username"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xsi:type="xs:anyType">testuser@123.org
  </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.Phone"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xsi:type="xs:anyType">415-123-1234
  </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.FirstName"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xsi:type="xs:anyType">Testuser
  </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.LanguageLocaleKey"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xsi:type="xs:anyType">en_US
  </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.CompanyName"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xsi:type="xs:anyType">Salesforce.com
  </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.Alias"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xsi:type="xs:anyType">tlee2
  </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.CommunityNickname"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xsi:type="xs:anyType">tlee2
  </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.UserRoleId"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xsi:type="xs:anyType">0000000000000000
  </saml:AttributeValue>
</saml:Attribute>
```

```

<saml:Attribute Name="User.Title"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xsi:type="xs:anyType">Mr.
  </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.LocaleSidKey"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xsi:type="xs:anyType">en_CA
  </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.Email"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xsi:type="xs:anyType">testuser@salesforce.com
  </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.FederationIdentifier"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xsi:type="xs:anyType">tlee2
  </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.TimeZoneSidKey"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xsi:type="xs:anyType">America/Los_Angeles
  </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.LastName"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xsi:type="xs:anyType">Lee
  </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.ProfileId"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xsi:type="xs:anyType">00ex0000001pBNL
  </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.IsActive"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xsi:type="xs:anyType">1
  </saml:AttributeValue>
</saml:Attribute>

<saml:Attribute Name="User.EmailEncodingKey"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xsi:type="xs:anyType">UTF-8
  </saml:AttributeValue>
</saml:Attribute>

</saml:AttributeStatement>

```

Reviewing the SAML Login History

User Permissions Needed	
To view the settings:	“View Setup and Configuration”
To edit the settings:	“Customize Application” AND “Modify All Data”

When a user logs in to Database.com from another application using single sign-on, SAML assertions are sent to the Database.com login page. The assertions are checked against assertions in the authentication certificate specified in **Security Controls > Single Sign-On Settings**. If a user fails to log in, a message is written to the login history log that indicates why the login failed. In addition, the [SAML Assertion Validator](#) may be automatically populated with the invalid assertion.

To view the login history, click **Users > Login History**. After viewing the login history, you may want to share the information with your identity provider.

The following are the possible failures:

Assertion Expired

An assertion's [timestamp](#) is more than five minutes old.



Note: Database.com does make an allowance of three minutes for clock skew. This means, in practice, that an assertion can be as much as eight minutes passed the timestamp time, or three minutes before it. This amount of time may be less if the assertion's validity period is less than five minutes.

Assertion Invalid

An assertion is not valid. For example, the `<Subject>` element of an assertion might be missing.

Audience Invalid

The value specified in `<Audience>` must be `https://saml.salesforce.com` or `https://saml.database.com`, depending on the Entity ID.

Configuration Error/Perm Disabled

Something is wrong with the SAML configuration in Database.com. For example, the uploaded certificate might be corrupted, or the organization preference might have been turned off. Check your configuration in **Security Controls > Single Sign-On Settings**, get a sample SAML assertion from your identity provider, and click [SAML Assertion Validator](#).

Issuer Mismatched

The issuer or entity ID specified in an assertion does not match the issuer specified in your Database.com configuration.

Recipient Mismatched

The recipient specified in an assertion does not match the recipient specified in your Database.com configuration.

Replay Detected

The same assertion ID was used more than once. [Assertion IDs](#) must be unique within an organization.

Signature Invalid

The signature in an assertion cannot be validated by the certificate in your Database.com configuration.

Subject Confirmation Error

The <Subject> specified in the assertion does not match the SAML configuration in Database.com.

Configuring SAML Settings for Single Sign-On

User Permissions Needed	
To view the settings:	“View Setup and Configuration”
To edit the settings:	“Customize Application” AND “Modify All Data”

From this page, you can configure your organization to use single sign-on. You can also set up just-in-time provisioning. Work with your identity provider to properly configure these settings. For more information about single sign-on, see [About Single Sign-On](#). For more information about just-in-time provisioning, see [About Just-In-Time Provisioning](#).

Configuring Single Sign-on

To configure SAML settings for single sign-on from your corporate identity provider to Database.com:

1. [Gather information from your identity provider](#).
2. [Provide information to your identity provider](#).
3. In Database.com, navigate to **Security Controls > Single Sign-On Settings**, and click **Edit**.
4. Select **SAML Enabled**. You must enable SAML to view the SAML single sign-on settings.
5. Specify the SAML version used by your identity provider.
6. Enter the **Issuer**. This is often referred to as the entity ID for the identity provider.
7. For the **Identity Provider Certificate**, use the **Browse** button to locate and upload the authentication certificate issued by your identity provider.
8. For SAML 2.0, if your identity provider has specific login or logout pages, specify them in **Identity Provider Login URL** and **Identity Provider Logout URL**, respectively.
9. For the **Custom Error URL**, specify the URL of the page users should be directed to if there's an error during SAML login. It must be a publicly accessible page. The URL can be absolute or relative.
10. If you are enabling just-in-time provisioning for security, check **User Provisioning Enabled**.

Note:

- Just-in-time provisioning requires a Federation ID in the user type. In **SAML User ID Type**, select Assertion contains the Federation ID from the User object.
- If your identity provider previously used the Database.com username, communicate to them that they must use the Federation ID.

11. For the **SAML User ID Type**, **SAML User ID Location**, and **other values as appropriate**, specify the value provided by your identity provider.
12. If your Database.com organization has **domains** deployed, specify whether you want to use the base domain (<https://saml.salesforce.com> or <https://saml.database.com>) or the custom domain for the **Entity ID**. You must share this information with your identity provider.



Tip: Generally, use the custom domain as the entity ID. If you already have single sign-on configured before deploying a domain, the base domain is the entity ID.

13. Click **Save**.

If your identity provider supports metadata, and if you've configured SAML using version 2.0, you can click **Download Metadata** to download an XML configuration file to send them, which they can then upload to automatically configure their settings for connecting to your Database.com organization.

After you have configured and saved your SAML settings, test them by trying to access the identity provider's application. Your identity provider directs the user's browser to POST a form containing SAML assertions to the Database.com login page. Each assertion is verified, and if successful, single sign-on is allowed.

If you have difficulty signing on using single sign-on after you have configured and saved your SAML settings, use the [SAML Assertion Validator](#). You may have to obtain a SAML assertion from your identity provider first.

If your users are having problems using SAML to login, you can [review the SAML login history](#) to determine why they were not able to log in and share that information with your identity provider.

If you are using SAML version 2.0, after you've finished configuring SAML, the OAuth 2.0 Token Endpoint field is populated. Use this with the [Web single sign-on authentication flow](#) on page 345 for OAuth 2.0.

Viewing Single Sign-On Settings

User Permissions Needed	
To view the settings:	“View Setup and Configuration”
To edit the settings:	“Customize Application” AND “Modify All Data”

After you have configured your Database.com organization to use SAML, you can view the single sign-on settings. Click **Security Controls > Single Sign-on Settings**.

This page lists the details of your SAML configuration. Most of these fields are the same as the fields on the page where you [configured SAML](#). The following fields contain information automatically generated by completing the configuration. The available fields depend on your configuration.

Field	Description
Database.com Login URL	For SAML 2.0 only. If you select “Assertion contains User's salesforce.com username” for SAML User ID Type and “User ID is in the NameIdentifier element of the Subject statement” for SAML User ID Location, this URL is the URL associated with login for the Web single sign-on OAuth assertion flow. See SAML Assertion Flow on page 345
Database.com Logout URL	For SAML 2.0. Displays the Database.com logout URL that the user is directed to after he or she logs off. This URL is only used if no value is specified for Identity Provider Logout URL.
OAuth 2.0 Token Endpoint	For SAML 2.0 only: The ACS URL used with enabling Database.com as an identity provider in the Web single sign-on OAuth assertion flow. See SAML Assertion Flow on page 345.

From this page you can do any of the following:

- Click **Edit** to change the existing SAML configuration.

- Click **SAML Assertion Validator** to validate the SAML settings for your organization using a SAML assertion provided by your identity provider.
- If your identity provider supports metadata, and if you've configured SAML using version 2.0, you can click **Download Metadata** to download an XML configuration file to send them, which they can then upload to automatically configure their settings for connecting to your Database.com organization.

Validating SAML Settings for Single Sign-On

User Permissions Needed	
To view the settings:	“View Setup and Configuration”
To edit the settings:	“Customize Application” AND “Modify All Data”

If your users have difficulty logging into Database.com after you [configure Database.com for single sign-on](#), use the SAML Assertion Validator and the [login history](#) to validate the SAML assertions sent by your identity provider.

- Obtain a SAML assertion from your identity provider. The assertion can be either in plain XML format or a base64 encoded.
- If a user tries to log in to Database.com and fails, the invalid SAML assertion is used to automatically populate the SAML Assertion Validator if possible.
- Click **Security Controls > Single Sign-On Settings**, then click **SAML Assertion Validator**.
- Enter the SAML assertion into the text box, and click **Validate**.
- Share the results of the [validation errors](#) with your identity provider.

SAML Assertion Validation Errors

User Permissions Needed	
To view the settings:	“View Setup and Configuration”
To edit the settings:	“Customize Application” AND “Modify All Data”

Database.com imposes the following validity requirements on assertions:

Authentication Statement

The identity provider must include an `<AuthenticationStatement>` in the assertion.

Conditions Statement

If the assertion contains a `<Conditions>` statement, it must contain a valid timestamp.

Timestamps

The validity period specified in an assertion is honored. In addition, an assertion's timestamp must be less than five minutes old, plus or minus three minutes, regardless of the assertion's validity period setting. This allows for differences between machines. The NotBefore and NotOnOrAfter constraints must also be defined and valid.

Attribute

If your Database.com configuration is set to User ID is in an Attribute element, the assertion from the identity provider must contain an <AttributeStatement>.

If you are using SAML 1.1, both <AttributeName> and <AttributeNamespace> are required as part of the <AttributeStatement>.

If you are using SAML 2.0, only <AttributeName> is required.

Format

The Format attribute of an <Issuer> statement must be set to "urn:oasis:names:tc:SAML:2.0:nameid-format:entity" or not set at all.

For example:

```
<saml:Issuer
Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">https://www.database.com</saml:Issuer>
```

The following example is also valid:

```
<saml:Issuer >https://www.database.com</saml:Issuer>
```

Issuer

The issuer specified in an assertion must match the issuer specified in Database.com.

Subject

The subject of the assertion must be resolved to be either the Database.com username or the Federation ID of the user.

Audience

The <Audience> value is required and must match the Entity ID from the single sign-on configuration. The default value can be <https://saml.salesforce.com> or <https://saml.database.com>, depending on your Entity ID.

Recipient

The recipient specified in an assertion must match either the Database.com login URL specified in the Database.com configuration or the OAuth 2.0 token endpoint. This is a required portion of the assertion and is always verified.

Signature

A valid signature must be included in the assertion. The signature must be created using the private key associated with the certificate that was provided in the SAML configuration.

Recipient

Verifies that the recipient and organization ID received in the assertion matches the expected recipient and organization ID, as specified in the single sign-on configuration. This is an optional portion of the assertion and is only verified if it's present. For example:

```
Recipient that we found in the assertion: http://aalbert-salesforce.com:8081/
?saml=02HKiPoin4zeKLPYxfj3twkPsNSJF3fxsH0Jnq4vVeQr3xNkIWmZC_IVk3
Recipient that we expected based on the Single Sign-On Settings page:
http://asmith.salesforce.com:8081/
?saml=EK03Almz90Cik_ig0L97.0BRme6mT4o6nzi0t_JROL6HLbdR1WVP5aQ05w
Organization Id that we expected: 00Dx0000000BQ01I
Organization Id that we found based on your assertion: 00D000000000062
```

Using Identity URLs

User Permissions Needed

To manage, create, edit, and delete OAuth applications: “Manage Remote Access”

In addition to the access token, an identity URL is also returned as part of a complete response, in the `id` parameter.

The identity URL is both a string that uniquely identifies a user, as well as a RESTful API that can be used to query (with a valid access token) for additional information about the user. Database.com returns basic personalization information about the user, as well as important endpoints that the client can talk to, such as photos for the user, and API endpoints it may access.

The format of the URL is: `https://login.database.com/ID/orgID/userID`, where `orgId` is the ID of the Database.com organization that the user belongs to, and `userID` is the Database.com user ID.

 **Note:** For [Test Database Overview](#), `login.database.com` is replaced with `test.database.com`.

The URL must always be HTTPS.

Identity URL Parameters

The following parameters can be used with the access token and identity URL. They are used in an authorization request header or in a request with the `oauth_token` parameter. For more details, see [Using the Access Token](#) on page 342.

Parameter	Description
Access token	See Using the Access Token on page 342.
Format	<p>This parameter is optional. Specify the format of the returned output. Valid values are:</p> <ul style="list-style-type: none"> • <code>urlencoded</code> • <code>json</code> • <code>xml</code> <p>Instead of using the <code>format</code> parameter, the client can also specify the returned format in an accept-request header using one of the following:</p> <ul style="list-style-type: none"> • <code>Accept: application/json</code> • <code>Accept: application/xml</code>

Parameter	Description
	<ul style="list-style-type: none"> Accept: application/x-www-form-urlencoded <p>Note the following:</p> <ul style="list-style-type: none"> Wildcard accept headers are allowed. */* is accepted and returns JSON. A list of values is also accepted and is checked left-to-right. For example: application/xml,application/json,application/html,*/* returns XML. The format parameter takes precedence over the accept request header.
Version	This parameter is optional. Specify a Web services API version number, or the literal string, latest. If this value isn't specified, the returned API URLs contains the literal value {version}, in place of the version number, for the client to do string replacement. If the value is specified as latest, the most recent API version is used.
PrettyPrint	This parameter is optional, and is only accepted in a header, not as a URL parameter. Specify the output to be better formatted. For example, use the following in a header: X-PrettyPrint :1. If this value isn't specified, the returned XML or JSON is optimized for size rather than readability.
Callback	This parameter is optional. Specify a valid JavaScript function name. This parameter is only used when the format is specified as JSON. The output is wrapped in this function name (JSONP.) For example, if a request to https://server/id/orgid/userid/ returns {"foo":"bar"}, a request to https://server/id/orgid/userid/?callback=ba returns baz {"foo":"bar"};.

Identity URL Response

After making a valid request, a **302 redirect** to an instance URL is returned. That subsequent request returns the following information in JSON format:

- `id`—The identity URL (the same URL that was queried)
- `asserted_user`—A boolean value, indicating whether the specified access token used was issued for this identity
- `user_id`—The Database.com user ID
- `username`—The Database.com username
- `organization_id`—The Database.com organization ID
- `nick_name`—The community nickname of the queried user
- `display_name`—The display name (full name) of the queried user
- `email`—The email address of the queried user
- `status`—The user's current Chatter status.

- ◊ `created_date`:xsd datetime value of the creation date of the last post by the user, for example, 2010-05-08T05:17:51.000Z
- ◊ `body`: the body of the post

- `photos`—A map of URLs to the user's profile pictures



Note: Accessing these URLs requires passing an access token. See [Using the Access Token](#) on page 342.

- ◊ `picture`
- ◊ `thumbnail`

- `urls`—A map containing various API endpoints that can be used with the specified user.



Note: Accessing the REST endpoints requires passing an access token. See [Using the Access Token](#) on page 342.

- ◊ `enterprise` (SOAP)
- ◊ `metadata` (SOAP)
- ◊ `partner` (SOAP)
- ◊ `profile`
- ◊ `feeds` (Chatter)
- ◊ `feed-items` (Chatter)
- ◊ `groups` (Chatter)
- ◊ `users` (Chatter)
- ◊ `custom_domain`—This value is omitted if the organization doesn't have a custom domain configured and propagated (see [My Domain Overview](#) on page 49)

- `active`—A boolean specifying whether the queried user is active
- `user_type`—The type of the queried user
- `language`—The queried user's language
- `locale`—The queried user's locale
- `utcOffset`—The offset from UTC of the timezone of the queried user, in milliseconds
- `last_modified_date`—xsd datetime format of last modification of the user, for example, 2010-06-28T20:54:09.000Z

The following is a response using XML format:

```
<?xml version="1.0" encoding="UTF-8"?>
<user xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<id>http://na1.salesforce.com/id/00Dx0000001T0zk/005x0000001S2b9</id>
<asserted_user>true</asserted_user>
<user_id>005x0000001S2b9</user_id>
<organization_id>00Dx0000001T0zk</organization_id>
<nick_name>admin1.2777578168398293E12foofoofoo</nick_name>
<display_name>Alan Van</display_name>
<email>admin@2060747062579699.com</email>
<status>
    <created_date xsi:nil="true"/>
    <body xsi:nil="true"/>
</status>
<photos>
    <picture>http://na1.salesforce.com/profilephoto/005/F</picture>
    <thumbnail>http://na1.salesforce.com/profilephoto/005/T</thumbnail>
</photos>
<urls>
```

```

<enterprise>http://na1.salesforce.com/services/Soap/c/{version}/00Dx0000001T0zk
</enterprise>
<metadata>http://na1.salesforce.com/services/Soap/m/{version}/00Dx0000001T0zk
</metadata>
<partner>http://na1.salesforce.com/services/Soap/u/{version}/00Dx0000001T0zk
</partner>
<rest>http://na1.salesforce.com/services/data/v{version}/
</rest>
<sobjects>http://na1.salesforce.com/services/data/v{version}/sobjects/
</sobjects>
<search>http://na1.salesforce.com/services/data/v{version}/search/
</search>
<query>http://na1.salesforce.com/services/data/v{version}/query/
</query>
<profile>http://na1.salesforce.com/005x0000001S2b9
</profile>
</urls>
<active>true</active>
<user_type>STANDARD</user_type>
<language>en_US</language>
<locale>en_US</locale>
<utcOffset>-28800000</utcOffset>
<last_modified_date>2010-06-28T20:54:09.000Z</last_modified_date>
</user>

```

The following is a response using JSON format:

```
{
  "id": "http://na1.salesforce.com/id/00Dx0000001T0zk/005x0000001S2b9",
  "asserted_user": true,
  "user_id": "005x0000001S2b9",
  "organization_id": "00Dx0000001T0zk",
  "nick_name": "admin1.2777578168398293E12foofoofoofoo",
  "display_name": "Alan Van",
  "email": "admin@2060747062579699.com",
  "status": {"created_date": null, "body": null},
  "photos": {"picture": "http://na1.salesforce.com/profilephoto/005/F",
    "thumbnail": "http://na1.salesforce.com/profilephoto/005/T"},
  "urls": {
    "enterprise": "http://na1.salesforce.com/services/Soap/c/{version}/00Dx0000001T0zk",
    "metadata": "http://na1.salesforce.com/services/Soap/m/{version}/00Dx0000001T0zk",
    "partner": "http://na1.salesforce.com/services/Soap/u/{version}/00Dx0000001T0zk",
    "rest": "http://na1.salesforce.com/services/data/v{version}/",
    "sobjects": "http://na1.salesforce.com/services/data/v{version}/sobjects/",
    "search": "http://na1.salesforce.com/services/data/v{version}/search/",
    "query": "http://na1.salesforce.com/services/data/v{version}/query/",
    "profile": "http://na1.salesforce.com/005x0000001S2b9"
  },
  "active": true,
  "user_type": "STANDARD",
  "language": "en_US",
  "locale": "en_US",
  "utcOffset": -28800000,
  "last_modified_date": "2010-06-28T20:54:09.000+0000"
}
```

After making an invalid request, the following are possible responses from Database.com:

Request Problem	Error Code
HTTP	403 (forbidden) — HTTPS_Required
Missing access token	403 (forbidden) — Missing_OAuth_Token
Invalid access token	403 (forbidden) — Bad_OAuth_Token
Users in a different organization	403 (forbidden) — Wrong_Org

Request Problem	Error Code
Invalid or bad user or organization ID	404 (not found) — Bad_Id
Deactivated user or inactive organization	404 (not found) — Inactive
User lacks proper access to organization	404 (not found) — No_Access or information
Request to the endpoint of a site	404 (not found) — No_Site_Endpoint
Invalid version	406 (not acceptable) — Invalid_Version
Invalid callback	406 (not acceptable) — Invalid_Callback

Understanding Just-in-Time Provisioning for SAML

About Just-in-Time Provisioning for SAML

With Just-in-Time provisioning, you can use a SAML assertion to create users on the fly the first time they try to log in. This eliminates the need to create user accounts in advance. For example, if you recently added an employee to your organization, you don't need to manually create the user in Database.com. When they log in with single sign-on, their account is automatically created for them, eliminating the time and effort with on-boarding the account. Just-in-Time provisioning works with your SAML identity provider to pass the correct user information to Database.com in a SAML 2.0 assertion. You can both create and modify accounts this way. Because Just-in-Time provisioning uses SAML to communicate, your organization must have SAML-based single sign-on enabled.

Benefits of Just-in-Time Provisioning

Implementing Just-in-Time provisioning can offer the following advantages to your organization.

- Reduced Administrative Costs:** Provisioning over SAML allows customers to create accounts on-demand, as part of the single sign-on process. This greatly simplifies the integration work required in scenarios where users need to be dynamically provisioned, by combining the provisioning and single sign-on processes into a single message.
- Increased Security:** Any password policies that you have established for your corporate network are also in effect for Database.com. In addition, sending an authentication credential that is only valid for a single use can increase security for users who have access to sensitive data.

Just-in-Time Provisioning Requirements

Just-in-Time provisioning requires the creation of a SAML assertion. Consider the following when creating your SAML assertion.

- Provision Version is supported as an optional attribute. If it isn't specified, the default is 1.0. For example:

```
<saml:Attribute Name="ProvisionVersion" NameFormat=
    "urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
    <saml:AttributeValue xsi:type="xs:anyType">1.0</saml:AttributeValue>
</saml:Attribute>
```

- ProfileIDs change per organization, even for standard profiles. To make it easier to find the profile name, Database.com allows you to do a profile name lookup by passing the ProfileName into the ProfileId field.

Field Requirements for the SAML Assertion

To correctly identify which object to create in Database.com, you must use the `User.` prefix for all fields passed in the SAML assertion. In this example, the `User.` prefix has been added to the `Username` field name.

```
<saml:Attribute
  Name="User.Username"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
  <saml:AttributeValue xsi:type="xs:anyType">testuser@123.org</saml:AttributeValue>
</saml:Attribute>
```

The following standard fields are supported.

Fields	Required	Comments
AboutMe		
Alias		If not present, a default is derived from FirstName and LastName.
CallCenter		
City		
CommunityNickname		If not present, a default is derived from the UserName.
CompanyName		
Country		
DefaultCurrencyIsoCode		Derived from organization settings.
DelegatedApproverId		
Department		
Division		
Email	Y	
EmailEncodingKey		If not present, a default is derived from the organization settings.
EmployeeNumber		
Extension		
Fax		
FederationIdentifier (insert only)		If present, it must match the SAML subject, or the SAML subject is taken instead. Can't be updated with SAML.
FirstName		
ForecastEnabled		
IsActive		
LastName	Y	
LanguageLocaleKey		
LocaleSidKey		If not present, a default is derived from the organization settings.
Manager		
MobilePhone		

Fields	Required	Comments
Phone		
ProfileId	Y	
ReceivesAdminInfoEmails		
ReceivesInfoEmails		
State		
Street		
TimeZoneSidKey		If not present, a default is derived from the organization settings.
Title		
Username (insert only)	Y	Can't update using SAML.
UserRoleId		Defaults to "no role" if blank.
Zip		

Other field requirements:

- Only text type custom fields are supported.
- Only the `insert` and `update` functions are supported for custom fields.
- When using the API for user creation, you can pass the new username into the `User.Username` field. You can also specify the `User.FederationIdentifier` if it is present. However, the `Username` and `FederationIdentifier` fields can't be updated with API.

Just-in-Time Provisioning Errors

This table shows the error codes for Just-in-Time provisioning for SAML. Errors are returned in the URL parameter, for example:

```
http://login.database.com/identity/jit/saml-error.jsp?
ErrorCode=5&ErrorDescription=Unable+to+create+user&ErrorDetails=
INVALID_OR_NULL_FOR_RESTRICTED_PICKLIST+TimeZoneSidKey
```



Note:

Database.com redirects the user to a custom error URL if one is specified in your SAML configuration. For more information on setting a custom error URL, see [Configuring SAML Settings for Single Sign-On](#). on page 278

Error Messages

Code	Description	Error Details
1	Missing Federation Identifier	MISSING_FEDERATION_ID
2	Mis-matched Federation Identifier	MISMATCH_FEDERATION_ID
3	Invalid organization ID	INVALID_ORG_ID
4	Unable to acquire lock	USER_CREATION_FAILED_ON_UROG
5	Unable to create user	USER_CREATION_API_ERROR

Code	Description	Error Details
6	Unable to establish admin context	ADMIN_CONTEXT_NOT_ESTABLISHED
8	Unrecognized custom field	UNRECOGNIZED_CUSTOM_FIELD
9	Unrecognized standard field	UNRECOGNIZED_STANDARD_FIELD
11	License limit exceeded	LICENSE_LIMIT_EXCEEDED
12	Federation ID and username do not match	MISMATCH_FEDERATION_ID_AND_USERNAME_ATTRS
13	Unsupported provision API version	UNSUPPORTED_VERSION
14	Username change isn't allowed	USER_NAME_CHANGE_NOT_ALLOWED
15	Custom field type isn't supported	UNSUPPORTED_CUSTOM_FIELD_TYPE
16	Unable to map an unique profile ID for the given profile name	PROFILE_NAME_LOOKUP_ERROR
17	Unable to map an unique role ID for the given role name	ROLE_NAME_LOOKUP_ERROR

Implementing Single Sign-On

About Single Sign-On

User Permissions Needed	
To view the settings:	“View Setup and Configuration”
To edit the settings:	“Customize Application” AND “Modify All Data”

Single sign-on is a process that allows network users to access all authorized network resources without having to log in separately to each resource. Single sign-on allows you to validate usernames and passwords against your corporate user database or other client application rather than having separate user passwords managed by Database.com.

Database.com offers the following ways to use single sign-on:

- **Federated authentication using Security Assertion Markup Language (SAML):** When federated authentication is enabled, Database.com does not validate a user's password. Instead, Database.com verifies an assertion in the HTTP POST request, and allows single sign-on if the assertion is true. This is the default form of single sign-on.
- **Delegated authentication:** When delegated authentication is enabled, Database.com does not validate a user's password. Instead, Database.com makes a Web services call to your organization to establish authentication credentials for the user. You must request that this feature be enabled by salesforce.com. Contact salesforce.com to enable delegated authentication single sign-on for your organization.

When you have an external identity provider, and configure single sign-on for your Database.com organization, Database.com is then acting as a *service provider*. You can also enable Database.com as an *identity provider*, and use single sign-on to connect to a different service provider. Only the service provider needs to configure single sign-on.

The Single Sign-On Settings page displays which version of single sign-on is available for your organization. To learn more about the single sign-on settings, see [Configuring SAML Settings for Single Sign-On](#) on page 278. For more information about SAML and Database.com security, see the [Security Implementation Guide](#).

Benefits of Single Sign-On

Implementing single sign-on can offer the following advantages to your organization:

- **Reduced Administrative Costs:** With single sign-on, users only need to memorize a single password to access both network resources or external applications and Database.com. When accessing Database.com from inside the corporate network, users are logged in seamlessly, without being prompted to enter a username or password. When accessing Database.com from outside the corporate network, the users' corporate network login works to log them in. With fewer passwords to manage, system administrators receive fewer requests to reset forgotten passwords.
- **Leverage Existing Investment:** Many companies use a central LDAP database to manage user identities. By delegating Database.com authentication to this system, when a user is removed from the LDAP system, they can no longer access Database.com. Consequently, users who leave the company automatically lose access to company data after their departure.
- **Time Savings:** On average, a user takes five to 20 seconds to log in to an online application; longer if they mistype their username or password and are prompted to reenter them. With single sign-on in place, the need to manually log in to Database.com is avoided. These saved seconds add up to increased productivity.
- **Increased Security:** Any password policies that you have established for your corporate network will also be in effect for Database.com. In addition, sending an authentication credential that is only valid for a single use can increase security for users who have access to sensitive data.

Enabling Single Sign-On

Database.com offers two ways to use single sign-on:

- **Delegated Authentication:** When delegated authentication is enabled, Database.com does not validate a user's password. Instead, Database.com makes a Web services call to your organization to establish authentication credentials for the user. You must request that this feature be enabled by salesforce.com. Contact salesforce.com to enable delegated authentication single sign-on for your organization. For more information, see "Understanding Delegated Authentication Single Sign-On" in the Database.com online help.
- **Federated Authentication:** When federated authentication is enabled, Database.com does not validate a user's password. Instead, Database.com verifies an assertion in the HTTP POST request, and allows single sign-on if the assertion is true. This is the default form of single sign-on. For more information, see "Configuring SAML Settings for Single Sign-On" in the Database.com online help.

Benefits of Single Sign-On

Implementing single sign-on can offer the following advantages to your organization:

- **Reduced Administrative Costs:** With single sign-on, users only need to memorize a single password to access both network resources or external applications and Database.com. When accessing Database.com from inside the corporate network, users are logged in seamlessly, without being prompted to enter a username or password. When accessing Database.com from outside the corporate network, the users' corporate network login works to log them in. With fewer passwords to manage, system administrators receive fewer requests to reset forgotten passwords.
- **Leverage Existing Investment:** Many companies use a central LDAP database to manage user identities. By delegating Database.com authentication to this system, when a user is removed from the LDAP system, they can no longer access Database.com. Consequently, users who leave the company automatically lose access to company data after their departure.
- **Time Savings:** On average, a user takes five to 20 seconds to log in to an online application; longer if they mistype their username or password and are prompted to reenter them. With single sign-on in place, the need to manually log in to Database.com is avoided. These saved seconds add up to increased productivity.

- **Increased Security:** Any password policies that you have established for your corporate network will also be in effect for Database.com. In addition, sending an authentication credential that is only valid for a single use can increase security for users who have access to sensitive data.

Delegated Authentication Best Practices

Consider the following best practices when implementing delegated authentication single sign-on for your organization.

- Your organization's implementation of the Web service must be accessible by Database.com servers. This means you must deploy the Web service on a server in your DMZ. Remember to use your server's external DNS name when entering the **Delegated Gateway URL** in the Delegated authentication section at **Security Controls > Single Sign-On Settings** in Database.com.
- If Database.com and your system cannot connect, or the request takes longer than 10 seconds to process, the login attempt fails. An error is reported to the user indicating that his or her corporate authentication service is down.
- Namespaces, element names, and capitalization must be exact in SOAP requests. Wherever possible, generate your server stub from the WSDL to ensure accuracy.
- For security reasons, you should make your Web service available by SSL only. You must use an SSL certificate from a trusted provider, such as Verisign or Thawte. For a full list of trusted providers, contact salesforce.com.
- The IP address that originated the login request is `sourceIp`. Use this information to restrict access based on the user's location. Note that the Database.com feature that validates login IP ranges continues to be in effect for single sign-on users. For more information, see [Setting Login Restrictions](#) on page 255.
- You may need to map your organization's internal usernames and Database.com usernames. If your organization does not follow a standard mapping, you may be able to extend your user database schema (for example, Active Directory) to include the Database.com username as an attribute of a user account. Your authentication service can then use this attribute to map back to a user account.
- We recommend that you do not enable single sign-on for system administrators. If your system administrators are single sign-on users and your single sign-on server has an outage, they have no way to log in to Database.com. System administrators should always be able to log in to Database.com so they can disable single sign-on in the event of a problem.

Federated Authentication using SAML Best Practices

Consider the following best practices when implementing federated single sign-on with SAML for your organization.

- Obtain the Recipient URL value from the configuration page and put it in the corresponding configuration parameter of your Identity Provider.
- Your identity provider must allow you to set the Service Provider's Audience URL, and it must be set to `https://saml.database.com`.
- Database.com allows a maximum of three minutes for clock skew with your IDP server; make sure your server's clock is up-to-date.
- If you are unable to log in with SAML assertion, always check the login history and note the error message.
- You need to map your organization's internal usernames and Database.com usernames. You have two choices to do this: add a unique identifier to the `FederationIdentifier` field of each Database.com user, or extend your user database schema (for example, Active Directory) to include the Database.com username as an attribute of a user account. Choose the corresponding option for the `SAML User ID Type` field and configure your authentication service to send the identifier in SAML assertions.
- Before allowing users to log in with SAML assertions, enable the SAML organization preference and provide all the necessary configurations.
- All test database copies are made with federated authentication with SAML disabled. Any configuration information is preserved, except the value for `Recipient URL` changes to `http://tapp0.database.com`. The `Recipient URL` is updated to match your test database URL, for example `http://cs1.database.com`, after you re-enable SAML. To enable SAML in the test database copy, click **Security Controls > Single Sign-On Settings**; then click **Edit**, and select `SAML Enabled`.

- Your identity provider must allow you to set the service provider's audience URL. The value must match the Entity ID value in the single sign-on configuration. The default value is <https://saml.database.com>.

Best Practices for Implementing Single Sign-On

User Permissions Needed	
To view the settings:	"View Setup and Configuration"
To edit the settings:	"Customize Application" AND "Modify All Data"

Database.com offers the following ways to use single sign-on:

- Federated authentication using Security Assertion Markup Language (SAML):** When federated authentication is enabled, Database.com does not validate a user's password. Instead, Database.com verifies an assertion in the HTTP POST request, and allows single sign-on if the assertion is true. This is the default form of single sign-on.
- Delegated authentication:** When delegated authentication is enabled, Database.com does not validate a user's password. Instead, Database.com makes a Web services call to your organization to establish authentication credentials for the user. You must request that this feature be enabled by salesforce.com. Contact salesforce.com to enable delegated authentication single sign-on for your organization.

Delegated Authentication Best Practices

Consider the following best practices when implementing delegated authentication single sign-on for your organization.

- Your organization's implementation of the Web service must be accessible by Database.com servers. This means you must deploy the Web service on a server in your DMZ. Remember to use your server's external DNS name when entering the Delegated Gateway URL in the Delegated authentication section at **Security Controls > Single Sign-On Settings** in Database.com.
- If Database.com and your system cannot connect, or the request takes longer than 10 seconds to process, the login attempt fails. An error is reported to the user indicating that his or her corporate authentication service is down.
- Namespaces, element names, and capitalization must be exact in SOAP requests. Wherever possible, generate your server stub from the WSDL to ensure accuracy.
- For security reasons, you should make your Web service available by SSL only. You must use an SSL certificate from a trusted provider, such as Verisign or Thawte. For a full list of trusted providers, contact salesforce.com.
- The IP address that originated the login request is sourceIp. Use this information to restrict access based on the user's location. Note that the Database.com feature that validates login IP ranges continues to be in effect for single sign-on users. For more information, see [Setting Login Restrictions](#) on page 255.
- You may need to map your organization's internal usernames and Database.com usernames. If your organization does not follow a standard mapping, you may be able to extend your user database schema (for example, Active Directory) to include the Database.com username as an attribute of a user account. Your authentication service can then use this attribute to map back to a user account.
- We recommend that you do not enable single sign-on for system administrators. If your system administrators are single sign-on users and your single sign-on server has an outage, they have no way to log in to Database.com. System administrators should always be able to log in to Database.com so they can disable single sign-on in the event of a problem.

Federated Authentication using SAML Best Practices

Consider the following best practices when implementing federated single sign-on with SAML for your organization.

- Obtain the Recipient URL value from the configuration page and put it in the corresponding configuration parameter of your Identity Provider.
- Your identity provider must allow you to set the Service Provider's Audience URL, and it must be set to `https://saml.database.com`.
- Database.com allows a maximum of three minutes for clock skew with your IDP server; make sure your server's clock is up-to-date.
- If you are unable to log in with SAML assertion, always check the login history and note the error message.
- You need to map your organization's internal usernames and Database.com usernames. You have two choices to do this: add a unique identifier to the `FederationIdentifier` field of each Database.com user, or extend your user database schema (for example, Active Directory) to include the Database.com username as an attribute of a user account. Choose the corresponding option for the `SAML User ID Type` field and configure your authentication service to send the identifier in SAML assertions.
- Before allowing users to log in with SAML assertions, enable the SAML organization preference and provide all the necessary configurations.
- All test database copies are made with federated authentication with SAML disabled. Any configuration information is preserved, except the value for `Recipient URL` changes to `http://tapp0.database.com`. The `Recipient URL` is updated to match your test database URL, for example `http://cs1.database.com`, after you re-enable SAML. To enable SAML in the test database copy, click **Security Controls > Single Sign-On Settings**; then click **Edit**, and select `SAML Enabled`.
- Your identity provider must allow you to set the service provider's audience URL. The value must match the `Entity ID` value in the single sign-on configuration. The default value is `https://saml.database.com`.

Understanding Delegated Authentication Single Sign-On

User Permissions Needed	
To view the settings:	“View Setup and Configuration”
To edit the settings:	“Customize Application” AND “Modify All Data”

Database.com uses the following process for authenticating users using delegated authentication single sign-on:

1. When a user tries to log in—either online or using the API—Database.com validates the username and checks the user's permissions and access settings.
2. If the user has the “Is Single Sign-On Enabled” user permission, then Database.com does not validate the username and password. Instead, a Web services call is made to the user's organization, asking it to validate the username and password.



Note: Database.com doesn't store, log, or view the password in any way. It is disposed of immediately once the process is complete.

3. The Web services call passes the `username`, `password`, and `sourceIp` to your Web service. (`sourceIp` is the IP address that originated the login request. You must create and deploy an implementation of the Web service that can be accessed by Database.com servers.)
4. Your implementation of the Web service validates the passed information and returns either `true` or `false`.
5. If the response is `true`, then the login process continues, a new session is generated, and the user proceeds to the application. If `false` is returned, then the user is informed that his or her username and password combination is invalid.



Note: System administrators should have single sign-on disabled. If your system administrators were single sign-on users and your single sign-on server had an outage, the administrators would have no way to log in to Database.com. System administrators should always be able to log in to Database.com so that they can disable single sign-on in the event of a problem.

Configuring Database.com for Delegated Authentication

User Permissions Needed	
To view the settings:	“View Setup and Configuration”
To edit the settings:	“Customize Application” AND “Modify All Data”

To enable delegated authentication single sign-on (SSO) for your organization:

1. Contact salesforce.com to enable delegated authentication single sign-on for your organization.
2. Build your single sign-on Web service:

- a. In Database.com, download the Web Services Description Language (WSDL) file, AuthenticationService.wsdl, by clicking **Develop > API > Download Delegated Authentication WSDL**. The WSDL describes the delegated authentication single sign-on service and can be used to automatically generate a server-side stub to which you can add your specific implementation. For example, in the WSDL2Java tool from Apache Axis, you can use the `--server-side` switch. In the wsdl.exe tool from .NET, you can use the `/server` switch.

For a sample request and response, see [Sample SOAP Message for Delegated Authentication](#) on page 274.

- b. Add a link to your corporate intranet or other internally-accessible site that takes the authenticated user's credentials and passes them through an HTTP POST to the Database.com login page.

Because Database.com does not use the `password` field other than to pass it back to you, you do not need to send a password in this field. Instead, you could pass another authentication token, such as a Kerberos Ticket so that your actual corporate passwords are not passed to or from Database.com.

You can configure the Database.com delegated authentication authority to allow only tokens or to accept either tokens or passwords. If the authority only accepts tokens, a Database.com user cannot log in to Database.com directly, because they cannot create a valid token. However, many companies choose to allow both tokens and passwords. In this environment, a user could still log in to Database.com through the login page.

When the Database.com server passes these credentials back to you in the `Authenticate` message, verify them, and the user will gain access to the application.

3. In Database.com, specify your organization's single sign-on gateway URL by clicking **Security Controls > Single Sign-On Settings > Edit**. Enter the URL in the **Delegated Gateway URL** text box.

For security reasons, Database.com restricts the outbound ports you may specify to one of the following:

- 80: This port only accepts HTTP connections.
- 443: This port only accepts HTTPS connections.
- 1024–66535 (inclusive): These ports accept HTTP or HTTPS connections.

4. Optionally, check the **Force Delegated Authentication Callout** box.



Note: When this box is unchecked, a call is not made to the SSO endpoint if the login attempt first fails because of login restrictions within the Database.com organization. If you must record every login attempt, then check this box to force a callout to the SSO endpoint regardless of login restriction failures.

5. Enable the “Is Single Sign-On Enabled” permission. For more information, see [Overview of User Permissions and Access](#) on page 207.



Important: If single sign-on is enabled for your organization, API users can't log into Database.com unless their IP address is included on your organization's list of trusted IP addresses or on their profile, if their profile has IP address restrictions set. Furthermore, the single sign-on authority usually handles login lockout policies for users with the “Is Single Sign-On Enabled” permission. However, if the security token is enabled for your organization, then your organization's login lockout settings determine the number of times a user can attempt to log in with an invalid security token before being locked out of Database.com. For more information, see [Setting Login Restrictions](#) on page 255. For information on how to view login errors, see [Viewing Single Sign-On Login Errors](#) on page 295.

Viewing Single Sign-On Login Errors

If your organization is enabled for Single Sign-On using delegated authentication and has built a Single Sign-On solution, you can view the most recent Single Sign-On login errors for your organization.

1. Click **Manage Users > Delegated Authentication Error History**.
2. For the twenty-one most recent login errors, you can view the user's username, login time, and the error.



Note: Contact salesforce.com to learn more about enabling Single Sign-On for your organization.

Managing Database.com Certificates and Keys

About Database.com Certificates and Keys

User Permissions Needed	
To create, edit, and manage certificates:	“Customize Application”

To work with Database.com certificates and keys, click **Security Controls > Certificate and Key Management**. From this page you can:

- Manage your certificates
- Manage your master encryption key

Certificates

Database.com certificates and key pairs are used for signatures that verify a request is coming from your organization. They are used for authenticated SSL communications with an external web site, or when using your organization as an Identity Provider. You only need to generate a Database.com certificate and key pair if you're working with an external website that wants verification that a request is coming from a Database.com organization.

Database.com offers two types of certificates:

Self-signed

A self-signed certificate is signed by Database.com. Not all external websites accept self-signed certificates.

CA-signed

A CA-signed certificate is signed by an external certificate authority (CA). Most external websites accept CA-signed certificates. You must first generate the certificate signing request to send to a CA, and then import the signed version of the certificate before you can use it.

Master Encryption Keys

Fields that are encrypted using encrypted custom fields, such as social security number or credit card number, use a master encryption key to encrypt the data. This key is automatically assigned when you enable encrypted fields for your organization. You can manage the master key based on your organization's security needs and regulatory requirements.

Creating Salesforce Certificates and Key Pairs

User Permissions Needed	
To create, edit, and manage certificates:	“Customize Application”

Database.com offers two types of certificates:

Self-signed

A self-signed certificate is signed by Database.com. Not all external websites accept self-signed certificates.

CA-signed

A CA-signed certificate is signed by an external certificate authority (CA). Most external websites accept CA-signed certificates. You must first generate the certificate signing request to send to a CA, and then import the signed version of the certificate before you can use it.

To create a Database.com certificate:

1. Go to **Security Controls > Certificate and Key Management**.
2. Select either **Create Self-Signed Certificate** or **Create CA-Signed Certificate**, based on what kind of certificate your external website accepts. You can't change the type of a certificate after you've created it.
3. Enter a descriptive label for the Database.com certificate. This name is used primarily by administrators when viewing certificates.
4. Enter the **Unique Name**. This name is automatically populated based on the certificate 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. Use the **Unique Name** when referring to the certificate using the Force.com Web services API or Apex.
5. Select a **Key Size** for your generated certificate and keys. We recommend that you use the default key size of 2048 for security reasons. Selecting 2048 generates a certificate using 2048-bit keys and is valid for two years. Selecting 1024 generates a certificate using 1024-bit keys and is valid for one year.



Note: Once you save a Database.com certificate, you can't change the key size.

6. If you're creating a CA-signed certificate, you must also enter the following information. These fields are joined together to generate a unique certificate.

Field	Description
Common Name	The fully qualified domain name of the company requesting the signed certificate. This is generally of the form: <code>http://www.mycompany.com</code> .
Email Address	The email address associated with this certificate.
Company	Either the legal name of your company, or your legal name.
Department	The branch of your company using the certificate, such as marketing or accounting.
City	The city where the company resides.
State	The state where the company resides.
Country Code	A two-letter code indicating the country where the company resides. For the United States, the value is <code>US</code> .

7. Click **Save**.

After you successfully save a Database.com certificate, the certificate and corresponding keys are automatically generated.

You can have a maximum of 50 certificates.

After you create a CA-signed certificate, you must [upload the signed certificate](#) before you can use it.



Note: After you create a CA-signed certificate and certificate request, the certificate is not active and you can't use it until it's been signed by a certificate authority and uploaded into your organization.

Uploading Certificate Authority (CA)-Signed Certificates

User Permissions Needed	
To create, edit, and manage certificates:	“Customize Application”

After you [create a CA-signed certificate](#), you must do the following before the certificate is active and you can use the certificate.

1. Click **Security Controls > Certificate and Key Management**, click the name of the certificate, then click **Download Certificate Signing Request**.
2. Send the certificate request to the certificate authority of your choice.
3. After the certificate authority sends back the signed certificate, click **Security Controls > Certificate and Key Management**, click the name of the certificate, then click **Upload Signed Certificate**.
4. Click **Browse** to locate the CA-signed certificate. The CA-signed certificate must match the certificate created in Database.com. If you try to upload a different CA-signed certificate, the upload fails.
5. Click **Save** to finish the upload process. Click **Cancel** at any time to not upload the certificate.

After you successfully upload the signed certificate, the status of the certificate is changed to **Active** and you can use CA-signed certificate.



Note: You can't delete a CA-signed certificate after you've uploaded the signed certificate.

Managing Database.com Master Encryption Keys

User Permissions Needed	
To create, edit, and manage certificates:	“Customize Application”

Fields that are encrypted using encrypted custom fields, such as social security number or credit card number, use a master encryption key to encrypt the data. This key is automatically assigned when you enable encrypted fields for your organization. You can manage the master key based on your organization's security needs and regulatory requirements. With master encryption keys, you can do the following:

- Archive the existing key and create a new key
- Export an existing key after it's been archived
- Delete an existing key
- Import an existing key after it's been deleted

Archiving and Creating New Keys

To archive your current key and create a new key:

1. Click **Security Controls > Certificate and Key Management**.
2. Click **Archive Current Key and Create New Key**.
3. A warning message displays letting you know you are changing keys. Click **OK**.
4. A new key is generated, assigned the next sequential number, and activated.

All new data is encrypted using the new key. Existing data continues to use the archived key until the data is modified and saved. Then data is encrypted using the new key.

After you archive a key, you can export or delete it.

Exporting Keys

You can export your keys to a back-up location for safe keeping. salesforce.com recommends you export a copy of any key before deleting it.

Exporting creates a text file with the encrypted key. You can import the key back into your organization at a later time.

Click **Export** next to the key you want to export.

Deleting Keys

Don't delete a key unless you're absolutely certain no data is currently encrypted using the key. After you delete a key, any data encrypted with that key can no longer be accessed. If you export the key before you delete it, you can import the key back into your organization.

To delete a key, click **Delete** next to the key you want to delete.

The date the key is deleted displays.

Importing Keys

If you have data associated with a deleted key, you can import an exported key back into your organization. Any data that was not accessible becomes accessible again.

Click **Import** next to the key you want to import.

Editing Database.com Certificates and Key Pairs

User Permissions Needed	
To create, edit, and manage certificates:	“Customize Application”

After you create a Database.com certificate, you can only change the **Label** and the **Unique Name**. You can't change the type, key size, and so on. The certificate and the keys aren't regenerated when you edit a Database.com certificate.



Caution: Apex and the Force.com Web services API use the **Unique Name** to access the certificate. Changing the **Unique Name** could cause your code to break.

To edit a Database.com certificate:

1. Go to **Security Controls > Certificate and Key Management**.
2. Click **Edit** next to the name of a Database.com certificate.
3. Make your changes, then click **Save**.

To delete a certificate, click **Del**. If a certificate is being used as part of the configuration of your identity provider, you cannot delete it.



Note: You can't delete a CA-signed certificate after you've uploaded the signed certificate.

Viewing Database.com Certificates and Key Pairs

User Permissions Needed	
To create, edit, and manage certificates:	“Customize Application”

To view the details of a Database.com certificate, click **Security Controls > Certificate and Key Management**, then click the name of a certificate.

From the certificate detail page, you can do any of the following:

- Click **Edit** to [edit the label or unique name](#) of the certificate.
- Click **Delete** to delete the certificate.



Note: You can't delete a CA-signed certificate after you've uploaded the signed certificate.

- Click **Download Certificate** to download the full Base-64 encoded certificate. This is only available for active certificates. For CA-signed certificates, you must first [upload the signed certificate](#) before you can download or use it.

- Click **Download Certificate Signing Request** for CA-signed certificates that have not yet had the signed certificate uploaded.
- Click **Upload Signed Certificate** to upload the CA-signed certificate.

Managing Identity Providers and Service Providers

About Identity Providers and Service Providers

User Permissions Needed
Define and modify identity providers and service providers: “Customize Application”

An *identity provider* is a trusted provider that enables you to use single sign-on to access other websites. A *service provider* is a website that hosts applications. You can enable Database.com as an identity provider, then define one or more service providers, so your users can access other applications directly from Database.com using single sign-on. This can be a great help to your users: instead of having to remember many passwords, they will only have to remember one.

Before you can enable Database.com as an identity provider, you must [set up a domain](#).

Enabling Database.com as an identity provider requires a [self-signed Database.com certificate and key pair](#). If you haven't generated a Database.com certificate and key pair, one is automatically created for you when you enable Database.com as an identity provider. You also have the option of picking an already generated certificate, or creating one yourself.

Database.com uses the SAML 2.0 standard for single sign-on and generates SAML assertions when configured as an identity provider.

Use the [identity provider error log](#) if your users have errors when trying to log into your service provider's apps.

Using Identity Providers and Service Providers

Database.com supports the following:

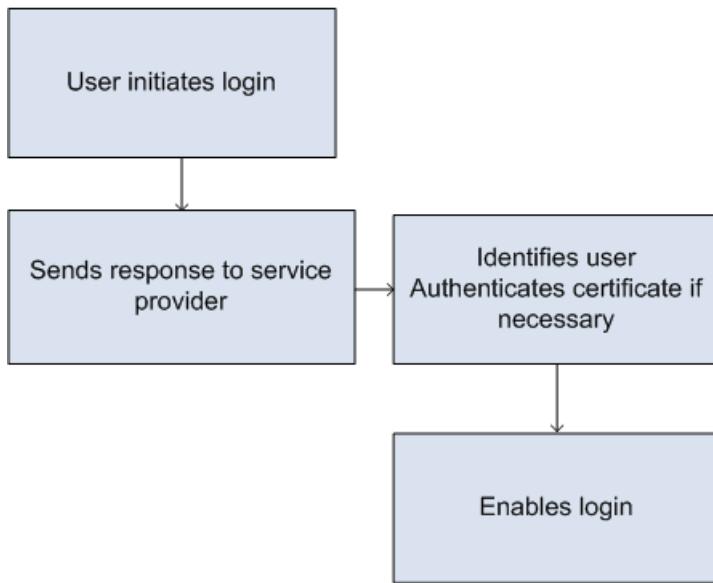
- Identity-provider-initiated login—when Database.com logs into a service provider at the initiation of the end-user
- Service-provider-initiated login—when the service provider requests Database.com to authenticate a user, at the initiation of the end-user

The following is the general flow when Database.com as an identity provider logs into a service provider.

1. User tries to access a service provider already defined in Database.com.
2. Database.com sends a [SAML response](#) to the service provider. The following is an example of a response (you might want to share this with your service provider):

Salesforce.com as Identity Provider

Service Provider



3. Service provider identifies the user and authenticates the certificate.
4. If the user is identified, they are logged into the service provider.

The following is the general flow when a service provider initiates login and uses Database.com to identify the user.

1. The service provider sends a valid SAML request. The endpoint is automatically generated when the service provider is defined—the SP-Initiated POST Endpoint.
2. Database.com identifies the user included in the SAML request.

```

<samlp:AuthnRequest ID="bndkmeemcaamihajeloilkagfdliilbhjjnmlmfo" Version="2.0"
IssueInstant="2010-05-24T22:57:19Z"
ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
ProviderName="google.com" IsPassive="false"
AssertionConsumerServiceURL="https://www.google.com/a/resp.info/acs">
<saml:Issuer>google.com</saml:Issuer>
<samlp:NameIDPolicy AllowCreate="true"
Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified"/>
</samlp:AuthnRequest>
  
```

If a certificate was included as part of the definition, Database.com authenticates the certificate.

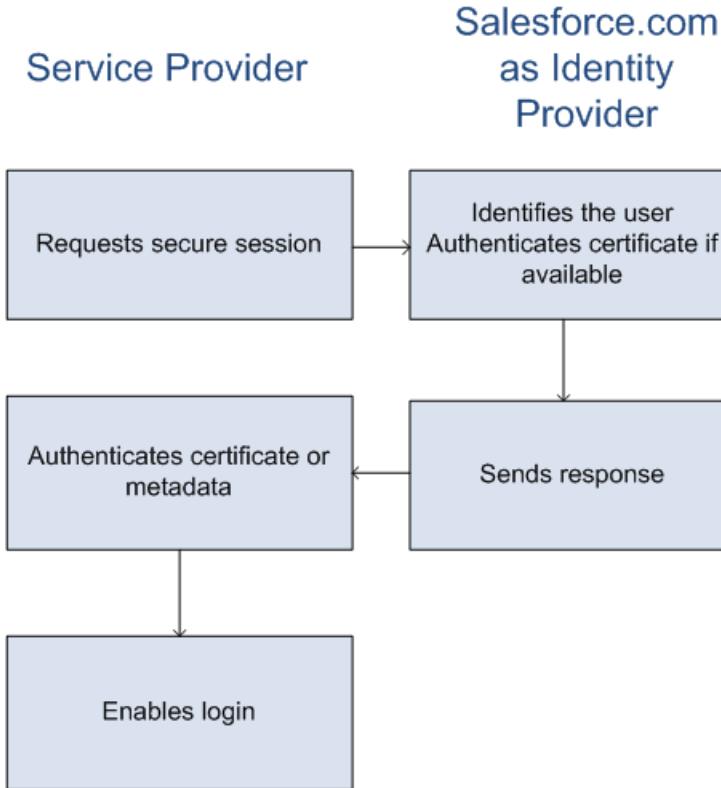


Note: If a certificate is included in the service provider definition, and the SAML request does not contain a certificate, the request fails, and the user is not logged in using Database.com. If the definition does not include a certificate, and the request includes one, the request succeeds if the user is identified correctly.

3. If the user isn't already logged into Database.com, they are prompted to do so.
4. Database.com sends a [SAML response](#) to the service provider.
5. The service provider authenticates the SAML response sent by Database.com. If the user has been authenticated, they are logged into the service provider. The user is also logged into Database.com.



Important: Database.com doesn't provide any mechanism for automatically logging the user out of Database.com when they log out of the service provider.



The following is an example of the SAML response from Database.com. You might want to share this information with your service provider.

```

<samlp:Response Destination="https://login-blitz03.soma.salesforce.com/
?saml=MgoTx78aEPa2r1BHKCHmlfUKhH2mkDrXOjmYcjHG_qNDbSRM_6ZAo.wvGk"
>ID="_0f551f9288c8b76f21c3d4d15c9cd1df1290476801091"
InResponseTo="2INwHuINDJlVjo8chM.Fpw_uLuKyj0MARVx2IJD569kZYLosBwuaSbzxxOPQjDtfw52tJB10VfgPw2p5g7N1v5k1QDzR0EJYGgn0d0z8CIiUOY31Y
Bdk7gwEkTygiK_1b46IO1fzBFoaRTzwvf1JN4qnkGttw3J6L4bopRI8hSQnCumM_Cvn3DHZVN.KtrzzOaf1cmFSCY.bj1wvruSGQCooTRSSQ"
IssueInstant="2010-11-23T01:46:41.091Z" Version="2.0">
<saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity"
>identityorg.blitz03.blitz.salesforce.com</saml:Issuer>
-
<ds:Signature>
-
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
-
<ds:Reference URI="#_0f551f9288c8b76f21c3d4d15c9cd1df1290476801091">
-
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
-
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
<ec:InclusiveNamespaces PrefixList="ds saml samlp xs"/>
```

```

</ds:Transform>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>4NVTbQ2WavD+ZBiYQ7ufc8EhtZw=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
-
<ds:SignatureValue>

eqrkFxNlJRCT4VQ7tt7wKZGK7oLCCCa4gV/HNcL03RoKbSXICwu2CAqW0qTSj25FqhRe2fOwAYa5
xFWat7Fw2bbncU+/nnuVNzut8HEEQoHiQA/Jrh7XB4CN1OpM1QRvgB5Dtckj/01I4h3X3TFix57B
sgZJGbb5PWEqSH3ZAl+NPvW9nNtYQ1FyCTe9+cw2BhCxFgSwfP3/kIYHSM2gbIy27CrRrFS11AqP
hKSLaH+ntH1E09gp78RSyJ2WKFGJU22sE9RJSZwdVw3VGG06Z6RpSjPJtaREELhhIBWTHNoF+VvJ
2Hbexjew6C0081XRDe8dbrrPIRK/qzHZYf1H0g==
</ds:SignatureValue>
-
<ds:KeyInfo>
-
<ds:X509Data>
-
<ds:X509Certificate>
MIIEbjCCA1agAwIBAgIOASH04Qu1AAAAAC1Xs7MwDQYJKoZIhvCNQEFBQAwfTEVMBMGA1UEAwM
SWRLbnRpdHkgT3JnMRgwFgYDVQQHDA1TYW4gRnJhbmNpc2NvMQswCQYDVQQIDAJDQTEEMMAoGA1UEBhMDVVNB
MB4XDTEwMDUwNzIyMjcwNvoXDTEyMDUwNjIyMjcwNVowfTEVMBMGA1UEAwMSWR1bnRpdHkgT3Jn
MRgwFgYDVQQHDA8wMEREMDAwMDAwMEZIOGwxFzAVBgnVBaoMD1NhbgVzZm9yY2UuY29tMRYwFAYD
VQQHDA1TYW4gRnJhbmNpc2NvMQswCQYDVQQIDAJDQTEEMMAoGA1UEBhMDVVNBMIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEAYM4/sjoaiZbnWTDjt9mGht2fDGxnLCWGMJ+D+9NWxD5wM15N
SFEcflpI9W4makcCGvoac+CVbPTmOUzOsCQzu7iGkLeMMpngf2Xql1nJg14ejuH8socNrDtltamk
hC08KAml3Wm/okllqsjVO18H5j2tbvm6HkvLVj2NDLRY6kUejVZMGjGwV5E0FJliwgIip4sCchl
dkahbNjbikiv1MASxHbtBt3wnKZWJq3Jts0valsaZUVmEwGD1VW43QPFOs7eV3IJFFhyCPV8yF
N3k0wCkCVBWoknwkMA8CbD+p6qNBVm vh3F3IaW2oym/1eSvtMLNtrPJeZssqDYqgQIDAQABo4Hr
MIHoMB0GA1UdDgQWBTTYSVEZ9r8Q8T2rbZxPFTPYpZKWITCBtQYDVR0jBIGtMIGqgBTYSVEZ9r8Q
8T2rbZxPFTPYpZKWIAgBgaR/MH0xFTATBgNVBAMMDE1kZW50aXR5IE9yZzEYMBYGA1UECwwPMDBE
RDAwMDAwMDBGSDhsMRcwFQYDVQQKDA5TYWx1c2ZvcmN1LmNvbTEWMBQGA1UEBwwNU2FuIEZyYW5j
aXNjbzELMAkGA1UECAwCQ0ExDDAKBgNVBAYTA1VTQYIOASH04QupAAAAAC1Xs7MwDwYDVR0TAQH/
BAUwAwEB/zANBgkqhkiG9w0BAQUFAAOCAQEAnA05Tqcc56E6Jv8itwjtBpvR+WHEMnZgQ9cCPF5Q
VACd5v7I/srx4ZJt/Z04RZkmX1FXla0M7JGU63eELHYG1DxT1SpGmpOL7xfBn7QUoh8Rmpp3BZC
WCPICVQHLS1LushsrpbWu+85tgz1VN4sFVB18F9rohhbM1dMOUAksQgM3avcZ2vkugKhX40vIuf
Gw4wXZe4TBCfQay+eDONYhYnmIxVV+dJyHheENOYfVqlau8RMNhRNmhX1GxXNQyU3kpMaTxOux8F
DyOjc5YPoe6PYQ0C/mC77ipnjJAjwm+Gw+heK/9NQ7fIonDOobbfu2rOmudtcKG74IDwkZL8HjA==
</ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</ds:Signature>
-
<samlp:Status>
<samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
</samlp:Status>
-
<saml:Assertion ID="e700bf9b25a5aebdb9495fe40332ef081290476801092"
IssueInstant="2010-11-23T01:46:41.092Z" Version="2.0">
<saml:Issuer
Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">identityorg.blitz03.blitz.salesforce.com</saml:Issuer>
-
<saml:Subject>
<saml:NameID
Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">charliemortimore@gmail.com</saml:NameID>
-
<saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
<saml:SubjectConfirmationData NotOnOrAfter="2010-11-23T01:51:41.093Z"
Recipient="https://login-blitz03.soma.salesforce.com/?saml=MgoTIX78aEPa2r1BHkCHmlfUKhH2mkDrXOjmYcjHG_qNDbsRM_6ZAo.wvGk"/>
</saml:SubjectConfirmation>
</saml:Subject>
-
<saml:Conditions NotBefore="2010-11-23T01:46:41.093Z" NotOnOrAfter="2010-11-23T01:51:41.093Z">
-
```

```

<saml:AudienceRestriction>
<saml:Audience>https://childorgb.blitz03.blitz.salesforce.com</saml:Audience>
</saml:AudienceRestriction>
</saml:Conditions>
-
<saml:AuthnStatement AuthnInstant="2010-11-23T01:46:41.092Z">
-
<saml:AuthnContext>
<saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified</saml:AuthnContextClassRef>
</saml:AuthnContext>
</saml:AuthnStatement>
-
<saml:AttributeStatement>
-
<saml:Attribute Name="userId"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
<saml:AttributeValue xsi:type="xs:anyType">005D0000001Ayzh</saml:AttributeValue>
</saml:Attribute>
-
<saml:Attribute Name="username"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
<saml:AttributeValue xsi:type="xs:anyType">admin@identity.org</saml:AttributeValue>
</saml:Attribute>
-
<saml:Attribute Name="email"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
<saml:AttributeValue xsi:type="xs:anyType">cmortimore@salesforce.com</saml:AttributeValue>
</saml:Attribute>
-
<saml:Attribute Name="is_portal_user"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
<saml:AttributeValue xsi:type="xs:anyType">false</saml:AttributeValue>
</saml:Attribute>
</saml:AttributeStatement>
</saml:Assertion>
</samlp:Response>

```

Identity Providers

An *identity provider* is a trusted provider that enables you to use single sign-on to access other websites. A *service provider* is a website that hosts applications. You can enable Database.com as an identity provider, then define one or more service providers, so your users can access other applications directly from Database.com using single sign-on. This can be a great help to your users: instead of having to remember many passwords, they will only have to remember one.

For more information, see “About Identity Providers and Service Providers” in the Database.com online help.

Enabling Database.com as an Identity Provider

User Permissions Needed
Define and modify identity providers and service providers: “Customize Application”

To enable Database.com as an identity provider:

1. Set up a domain.
2. Click **Security Controls > Identity Provider** and click **Enable Identity Provider**.
3. If you haven't created a self-signed certificate, one is automatically generated for you and assigned as the certificate for your identity provider. If you've already created self-signed certificates, select the certificate to use when securely communicating with other services. You can only use self-signed certificates for your identity provider. You can't use

CA-signed certificates. If you'd like to create one, select **Create a new certificate....** After you create the certificate, click **Security Controls > Identity Provider**, click **Enable Identity Provider**, and select the certificate you just created.

4. Click **Save**.

After you enable Database.com as an identity provider, you can [define service providers](#).

Viewing Your Identity Provider Details

User Permissions Needed

Define and modify identity providers and service providers: “Customize Application”

After you enable an identity provider for your organization, you can view the details by clicking **Security Controls > Identity Provider**. You might need to share this information, such as **Issuer**, with your service provider.

From this page you can click:

- **Edit** to change the certificate associated with your identity provider.



Caution: Changing the certificate can disable access to external applications. You might need to update all external applications to validate the new certificate information.

- **Disable** to disable your identity provider.



Caution: If you disable your identity provider, users can no longer access any external applications.

- **Download Certificate** to download the certificate associated with your identity provider. Your service provider can use this information for connecting to Database.com.
- **Download Metadata** to download the metadata associated with your identity provider. Your service provider can use this information for connecting to Database.com.
- In the service providers section, click **New** to define a new service provider. Next to the name of an already-defined service provider, click **Edit** to change its definition, click **Profiles** to add or remove user profiles that have access to this service provider, or click **Del** to delete it.

Prerequisites for Defining Service Providers

User Permissions Needed

Define and modify identity providers and service providers: “Customize Application”

Before defining a service provider:

1. [Enable Database.com as an identity provider](#).
2. Give your service provider information about your configuration of Database.com as an identity provider. This information is available as metadata that you can download and give to your service provider for easy configuration. However, not all service providers support metadata. If your service provider supports certificates instead, you might be required to download the certificate. Click **Security Controls > Identity Provider**, then click **Download Certificate** or **Download Metadata**.
3. Get the following information from your service provider:

- Assertion consumer service (ACS) URL
- Entity ID
- Subject type—specifies if the subject for the SAML response from Database.com (as an identity provider) is a Database.com user name or a federation ID
- Security certificate—only required when the service provider is initiating login to Database.com and signing their SAML requests

Defining Service Providers

User Permissions Needed

Define and modify identity providers and service providers: “Customize Application”

To define a service provider:

1. Complete the [prerequisites](#).
2. Click **Security Controls > Identity Provider**, then click **New** in the Service Providers section.
3. Specify a name for the service provider's application. This name is used in the profile settings.
4. Enter the assertion consumer service (ACS) URL. This value comes from the service provider.
5. Enter the entity ID. This value comes from the service provider.

If you are accessing multiple apps from your service provider, you only need to define the service provider once, and then use the `RelayState` parameter to append the URL values to direct the user to the correct app after signing in.



Important: Each entity ID must be unique in your organization.

6. Select the subject type. This value comes from the service provider.



Note: If the subject type is `Federation ID`, you must also [map the Database.com user to the app user](#).

7. Click **Service Provider Certificate** if the service provider gave you a security certificate. Browse your system for the certificate. This is only necessary if you plan to initiate logging into Database.com from the service provider and the service provider signs their SAML requests.



Important: If you upload a certificate, all SAML requests must be signed. If no certificate is uploaded, all SAML requests are accepted.

8. Click **Save**.

9. Select which profiles have access to this service provider.



Note: Permission sets can also be used to grant access to service providers.

If you click the checkbox at the top of the list, all profiles are selected.

10. Click **Save**.

Mapping Database.com Users to App Users

User Permissions Needed

Define and modify identity providers and service providers: “Customize Application”

If the Subject Type for the service provider definition is Federation ID, you must map the Database.com user to the username used to sign into the service provider.

To map a Database.com user to the app user:

1. Click **Manage Users > Users**, then click **Edit** for every user who needs to be mapped.
2. In Federation ID, under Single Sign On Information, enter the username to be used to log into the service provider.
3. Click **Save**.



Tip: Use the Web services API if you have a large number of user profiles or permission sets to update. See the [Web Services API Developer's Guide](#).

Viewing Your Service Provider Details

User Permissions Needed

Define and modify identity providers and service providers: “Customize Application”

After you define a service provider for your organization, you can view the details by clicking **Security Controls > Identity Provider**, then the name of the service provider. You might need to share this information, such as SP-Initiated POST Endpoint or SP-Initiated Redirect Endpoint, with your service providers.

From this page you can click:

- **Edit** to change the values of the service provider definition.
- **Delete** to delete a service provider definition.



Caution: If you delete a service provider definition, your users will no longer have access to that service provider.

- **Profile Access** to change which profiles have access to this service provider.

Using the Identity Provider Error Log

User Permissions Needed

Define and modify identity providers and service providers: “Customize Application”

The identity provider error log records problems with inbound SAML authentication requests from another app provider, and outbound SAML responses when Database.com is acting as an identity provider. To view the identity provider error log, click **Manage Users > Identity Provider Error Log**.

Examples Using Identity Providers and Service Providers

User Permissions Needed

Define and modify identity providers and service providers: “Customize Application”

This section contains two examples of setting up Database.com as an identity provider, then setting up two different service providers:

- [Google Apps](#)—shows service-provider initiated login.
- [Database.com](#)—shows identity-provider initiated login.

Setting up Single Sign-on to Google Apps Example

This example shows how to set up single sign-on from Database.com to Google Apps. In this example, Google is the service provider, and Google Apps is the app provided by the service provider.

For this example to work:

- You must already have a Premier Edition Google Apps account
- Your Database.com organization must be set up for single sign-on using SAML 2.0

The general steps are as follows, with more specifics on each step below.

1. [Generate a domain name and enable an identity provider](#) in your Database.com organization.
2. [Define the service provider in Database.com](#).
3. [Enable the Database.com user and profile](#).
4. [Setup Google Apps](#).
5. [Test your implementation](#).

Generating a Domain Name and Enabling an Identity Provider

To prepare your Database.com organization for this example, generate a domain name and enable Database.com as an identity provider:

1. Log into Database.com.
2. Generate a domain name for your organization:
 - a. Click **Company Profile > My Domain**, enter a new subdomain name, and click **Check Availability**.
 - b. If the name is available, click the **Terms and Conditions** check box, then click **Register Domain**.



Important: You must deploy your domain name before you can enable Database.com as an identity provider.

3. Enable Database.com as an identity provider:
 - a. Click **Security Controls > Identity Provider**.
 - b. Click **Enable**.
 - c. Click **Download Certificate**. Remember where you save the certificate, as you will upload it later.

Defining a Service Provider

To define the service provider:

1. Log into Database.com.

2. Click **Security Controls > Identity Provider**.
3. Click **New** in the Service Provider section and enter the following information:

Field	Value
Name	Google Apps
ACS URL	The URL for your Google App account, such as https://www.google.com/a/respond.info
Entity ID	google.com
Subject Type	Federation ID

4. Click **Save**.
5. Select the profiles allowed to access this service provider. You must select the current user's profile for this example to work.
6. Click **Save**.
7. Copy down the value in the **SP-Initiated Redirect Endpoint** field. You will use this value later.

Mapping the Database.com user to the Google Apps user

To map the Database.com user to the Google Apps user:

1. Click **My Personal Information > Personal Information**, then click **Edit**.
2. For **Federation ID**, enter the username you use to sign into Google Apps, for example, JSmith@TGroup.com.
3. Click **Save**.

Setting up Google Apps

To set up your Google Apps account:

1. Log into your Google Apps account.
2. Click the **Advanced tools** tab, then the **Set up single sign-on (SSO)** link.
3. Check the **Enable Single Sign-on** checkbox.
4. For **Sign-in page URL**, enter the URL copied from the **SP-Initiated Redirect Endpoint** field, from [defining a service provider](#).
5. For **Sign-out page URL**, specify the URL where you want your users to go after they log out of Google Apps, such as, <http://www.mydomain.salesforce.com>.
6. For **Change password URL**, use the following URL:
https://mydomain.salesforce.com/_ui/system/security/ChangePassword, where *mydomain* is the name you specified for your custom domain when you generated your domain.
7. For **Verification certificate**, upload the certificate you downloaded from [enabling an identity provider](#).
8. Click **Save Changes**.

Testing your Implementation

To verify that your Database.com organization can use single sign-on to Google Apps:

1. Log out of Google Apps and Database.com.
2. Try to access a Google app page, such as <http://docs.google.com/a/respond.info/> or <http://mail.google.com/a/respond.info/>.
3. You are redirected to a Database.com sign-on page. After you login, you are at the specified Google app page.

Setting up Single Sign-on From Database.com to Database.com

This example shows how to set up a Database.com app to initiate single sign-on from one Database.com organization to another.

The initiating Database.com organization, that is, the organization that you want to initially log into, acts as the *identity provider*. The Database.com organization that you want to access using an app acts as the *service provider*. For example, suppose you have two Database.com organizations: a sales organization and an ideas organization. You can set up single sign-on between the two organizations so your users only have to log into and remember the password for one.

For this example to work, your initiating Database.com organization must be set up for single sign-on using SAML 2.0. The general steps are as follows, with more specifics on each step below.

1. [Generate a domain name and enable an identity provider](#) in the Database.com organization that is acting as an identity provider.
2. [Set up the Database.com organization](#) that is acting as a service provider.
3. [Define the service provider app](#) in the Database.com organization that is acting as an identity provider.

Generating a Domain Name and Enabling an Identity Provider

All of the work in the following steps is done on the Database.com organization that is acting as the identity provider.

To prepare your Database.com organization for this example, generate a domain name and enable Database.com as an identity provider:

1. Log into Database.com.
2. Generate a domain name for your organization:
 - a. Click **Company Profile > My Domain**, enter a new subdomain name, and click **Check Availability**.
 - b. If the name is available, click the **Terms and Conditions** check box, then click **Register Domain**.



Important: You must deploy your domain name before you can enable Database.com as an identity provider.

3. Enable Database.com as an identity provider:
 - a. Click **Security Controls > Identity Provider**.
 - b. Click **Enable**.
 - c. Click **Download Certificate**. Remember where you save the certificate, as you will upload it later.

Setting up a Database.com organization as Service Provider

To configure a second Database.com organization as the service provider:

1. Log into the Database.com organization that acts as the service provider.
2. Enable and configure SAML:
 - a. Click **Security Controls > Single Sign-On Settings**, then click **Edit**.
 - b. Select the **SAML Enabled** check box.
 - c. Use the following settings:

Field	Value
SAML Version	2.0
Issuer	https://saml.salesforce.com

Field	Value
Identity Provider Certificate	Browse for the certificate you downloaded in enabling an identity provider .
SAML User ID Type	Select Assertion contains the Federation ID from the User object
SAML User ID Location	Select User ID is in the NameIdentifier element of the Subject statement

- d. Click **Save**.
 - e. Copy and save the values from the fields Database.com Login URL and Entity ID. You need these values later, when defining the Database.com service provider.
3. Link your user in the service provider organization to the user in the identity provider organization:
- a. Click **My Personal Information > Personal Information**, and click **Edit**.
 - b. For Federation ID, enter the username used to sign into the Database.com identity provider organization, for example, IDP_org@TGroup.com.
 - c. Click **Save**.

Defining the Service Provider in the Identity Provider Organization

To define the service provider:

1. Log into the Database.com organization that acts as the identity provider.
2. Click **Security Controls > Identity Provider**, then in the Service Provider section, click **New**.
3. Specify the following information:

Field	Value
Name	Database.com Service Provider
ACS URL	Use the Database.com Login URL from setting up the service provider
Entity Id	Use the Entity ID from setting up the service provider
Subject Type	Select Username

4. Click **Save**.
5. Select the profiles allowed to access this service provider. You must select the current user's profile for this example to work.
6. Click **Save**.
7. Copy down the value of the IdP-Initiated Login URL field. You will use this value later, in testing.

Managing Administrative Duties Monitoring Setup Changes

User Permissions Needed	
To view audit trail history:	"View Setup and Configuration"

The setup audit trail history helps you track the recent setup changes that you and other administrators have made to your organization. This can be especially useful in organizations with multiple administrators.

To view the setup audit trail history, click **Security Controls > View Setup Audit Trail**. To download your organization's full setup history for the past 180 days, click the **Download** link.

The setup audit trail history shows you the 20 most recent setup changes made to your organization. It lists the date of the change, who made it, and what the change was. Additionally, if a delegate makes a setup change on behalf of an end-user, the Delegate User column shows the delegate's username. For example, if a user grants login access to an administrator and the administrator makes a setup change, the administrator's username is listed.

The setup audit trail history tracks the following types of changes:

Setup	Changes Tracked
Administration	<ul style="list-style-type: none"> Company information, default settings such as language or locale, and company message changes Multiple currency setup changes User, role, permission set, and profile changes Email address changes for any user Adding or deleting certificates Domain name changes Enabling or disabling Database.com as an identity provider
Customization	<ul style="list-style-type: none"> Custom field and field-level security changes, including changes to formulas, picklist values, and custom field attributes like the format of auto-number fields or masking of encrypted fields Any changes made by salesforce.com Customer Support at your request Changes to custom objects Changes to field tracking in feeds
Security and Sharing	<ul style="list-style-type: none"> Public groups, sharing rule changes, and organization-wide sharing, including the Grant Access Using Hierarchies option Password policy changes Session settings changes, such as changing the session timeout setting Changes to delegated administration groups and the items delegated administrators can manage. Setup changes made by delegated administrators are tracked as well. How many records a user emptied from their Recycle Bin and from the organization's Recycle Bin Changes to SAML (Security Assertion Markup Language) configuration settings Changes to Database.com certificates Enabling or disabling identity providers Changes to service providers

Setup	Changes Tracked
Data Management	<ul style="list-style-type: none"> Mass delete use, including when a mass delete exceeds the user's Recycle Bin limit of 5000 deleted records. The oldest, excess records will be permanently removed from the Recycle Bin within two hours of the mass delete transaction time. Data export requests
Development	<ul style="list-style-type: none"> Changes to Apex classes and triggers Changes to custom settings Changes to remote access definitions
Various Setup	<ul style="list-style-type: none"> Creation of an API usage metering notification Changes to Workflow & Approvals settings Creation and deletion of workflow actions

Configuring Remote Settings

User Permissions Needed	
To configure remote settings:	“Modify All Data”

Before any Apex callout or JavaScript code can call an external site that site must be registered in the Remote Site Settings page, or the call will fail.

To access the page, click **Security Controls > Remote Site Settings**. This page displays a list of any remote sites already registered and provides additional information about each site, including remote site name and URL.

For security reasons, Database.com restricts the outbound ports you may specify to one of the following:

- 80: This port only accepts HTTP connections.
- 443: This port only accepts HTTPS connections.
- 1024–66535 (inclusive): These ports accept HTTP or HTTPS connections.

To register a new site:

- Click **New Remote Site**.
- Enter a descriptive term for the **Remote Site Name**.
- Enter the URL for the remote site.
- To allow access to the remote site regardless of whether the user's connection is over HTTP or HTTPS, select the **Disable Protocol Security** checkbox. When selected, Database.com can pass data from an HTTPS session to an HTTP session, and vice versa. Only select this checkbox if you understand the security implications.
- Optionally, enter a description of the site.
- Click **Save** to finish, or click **Save & New** to save your work and begin registering an additional site.

Managing Remote Access

Remote Access Application Overview

User Permissions Needed	
To manage, create, edit, and delete OAuth applications:	“Manage Remote Access”

A remote access application is an application external to Database.com that uses the OAuth protocol to verify both the Database.com user and the external application. OAuth is an open protocol that allows secure authentication for access to a user's data, without handing out the user's username and password. It is often described as the valet key of software access: a valet key only allows access to certain features of your car: you cannot open the trunk or glove compartment using a valet key.

The following is the general flow for using a remote access application with Database.com:

1. A developer uses the remote access pages in Database.com (**Develop > Remote Access**) to define a remote access application.
In this example, the remote access application is a web application, which uses data that already exists in Database.com.
2. The developer uses the generated client credentials from the remote access application detail page and develops their web application using an OAuth library.
3. A user starts to use the developer's web application and performs an action that requires access to their Database.com data.
4. The user is redirected to Database.com using the OAuth protocol, and presented with the standard Database.com login page.
5. Once the user successfully logs in, the Remote Access Authorization page displays. The user must verify if they want to grant the web application access to their Database.com data.
6. If the user approves access, they are redirected back to the originating web application with an authorization code.
7. The web application exchanges this code for an access token, which grants them access to the user's Database.com data.
In addition, depending on the authentication flow used, a refresh token might be granted, allowing continued access to the user's account.
8. After a user has granted access to a remote access application, he or she can revoke that access by clicking **My Personal Information > Personal Information** and clicking **Deny** next to the name of the application in the Remote Access related list.

While this example illustrates a common use of OAuth, Database.com supports a number of authentication flows for OAuth 2.0, so you can authenticate users of Web, JavaScript, desktop, or mobile applications. Database.com currently supports OAuth versions 1.0.A and 2.0.



Note: Database.com implemented draft 10 of the OAuth protocol from the IETF working group.

For more information on the OAuth standard, see the [OAuth.net documentation](#).

For more information on terminology, see [Remote Access Applications and OAuth Terminology](#) on page 315.



Note:

- OAuth does not automatically limit access to a user's data. Limits to data access are specified by the user's permissions.
- Users can authorize a remote access application to access their Database.com more than once, for example, for both a laptop and a desktop computer. The default limit is five per application per user. If a user tries to grant access to an application more than the organization limit, the access token for that application that hasn't been used for the longest period of time is revoked. Newer applications (using the OAuth 2.0 protocol) are automatically approved for additional devices after the user has granted access once.

Remote Access Applications and OAuth Terminology

User Permissions Needed
To manage, create, edit, and delete OAuth applications: “Manage Remote Access”

Access Token

A value used by the consumer to gain access to protected resources on behalf of the user, instead of using the user's Database.com credentials.

For OAuth 1.0.A, the access token must be exchanged for a session ID.

For OAuth 2.0, the access token is a session ID, and can be used directly.

Authorization Code

Only used in OAuth 2.0. A short-lived token that represents the access granted by the end user. The authorization code is used to obtain an access token and a refresh token. For OAuth 1.0.A, see [RequestToken](#).

Consumer

A Web site or application that uses OAuth to authenticate both the Database.com user as well as the application on the user's behalf.

Consumer Key

A value used by the consumer to identify itself to Database.com.

Consumer Secret

A secret used by the consumer to establish ownership of the consumer key.

Nonce

A number, often a random number, used during authentication to ensure that requests cannot be reused.

OAuth Protocol Parameters

Parameters with names beginning with `oauth_`, such as `oauth_consumer_key` and `oauth_nonce`.

Refresh Token

Only used in OAuth 2.0. A token used by the consumer to obtain a new access token, without having the end user approve the access again.

Request Token

A value used by the consumer to obtain authorization from the user, and exchanged for an access token. Request tokens are only used in OAuth 1.0.A. For OAuth 2.0, see [Authorization Code](#).

Service Provider

A Web application that allows access using OAuth. This is your Database.com instance after remote access has been enabled.

Token Secret

A secret used by the consumer to establish ownership of a given token, both for request tokens and access tokens.

User

An individual who has a Database.com login.

Getting Started with Remote Access Applications

User Permissions Needed
To manage, create, edit, and delete OAuth applications: “Manage Remote Access”

Before you start to use remote access applications, you need to consider the following:

- The version of the OAuth protocol to use.
- The type of application your developer is building. This determines which authentication flow to use.

Once you have made these decisions, you can define your remote access application.

Choosing a Version of the OAuth Protocol

Salesforce.com recommends that developers use the OAuth 2.0 protocol. OAuth 2.0 places a strong emphasis on developer simplicity, and you may find it easier to develop and integrate applications.

If you have an existing client or platform that leverages OAuth 1.0.A, it may be less work to use that version than to re-engineer it to use OAuth 2.0.

Determining Which Authentication Flow to Use

OAuth 1.0.A only supports a single authentication flow. See [OAuth 1.0.A Authentication Flow](#).

OAuth 2.0 supports a number of different application flows. These are dependent on the type of client you are developing.

- Web service—[OAuth 2.0 Web Server Authentication Flow](#)
- Mobile or desktop application, or JavaScript—[OAuth 2.0 User-Agent Flow](#)

CAPTCHA Security for Data Exports

By request, Database.com can also require users to pass a user verification test to export data from Database.com. This simple, text-entry test helps prevent malicious programs from accessing your organization's data, as well as reducing the risk of automated attacks. CAPTCHA is a type of network-based security. To pass the test, users must type two words displayed on an overlay into the overlay's text box field, and click a **Submit** button. Database.com uses CAPTCHA technology provided by [reCaptcha](#) to verify that a person, as opposed to an automated program, has correctly entered the text into the overlay. CAPTCHA stands for “Completely Automated Public Turing Test To Tell Computers and Humans Apart.”

Managing Remote Access Applications

Managing Your Remote Access Applications

User Permissions Needed

To manage, create, edit, and delete OAuth applications: “Manage Remote Access”

A remote access application is an application external to Database.com that uses the OAuth protocol to verify both the Database.com user and the external application. All remote access applications have been integrated with Database.com, such that they can access a subset of your Database.com data once you explicitly grant each application permission.

All remote access applications that have permission to access your Database.com data are listed in your personal information. Click **My Personal Information** > **Personal Information**, and go to the Remote Access section. From there you can do the following:

- View information about each remote access application that you have granted access to, as well as the number of times and the last time the application attempted to access your information.

Note:

- ◊ An application may be listed more than once. Each time you grant access to an application, it obtains a new access token. You must grant access to your Database.com data from each device that you use, for example, from both a laptop and a desktop computer. The default limit is five access tokens for each application. Newer applications (using the OAuth 2.0 protocol) are automatically approved for additional devices after you've granted access once. OAuth 2.0 applications are only listed once.
- ◊ Even if the remote access application tried and failed to access your information because it could not login, the Use Count and Last Used fields are still updated.

- Click **Revoke** to revoke the remote access application. After you revoke the application, the application can no longer use that particular remote access authorization token to access your Database.com data.

 **Important:** You must revoke all access tokens for a particular application to prevent it from accessing your Database.com data.

Deleting Remote Access Applications

User Permissions Needed

To manage, create, edit, and delete OAuth applications: “Manage Remote Access”

Click **Delete** to delete the remote access application. Click **Cancel** to go back to the previous page.

Viewing Remote Access Application Details

User Permissions Needed

To manage, create, edit, and delete OAuth applications: “Manage Remote Access”

Use the remote access pages to specify remote access applications that can access a Database.com instance.

To view the details for a defined application, click **Develop** > **Remote Access**, and click the name of the application.

From this page you can do any of the following:

- **Edit**—Change the name, contact information, and so on.



Note: Even if you change the name of the application, the consumer key and consumer secret are **not** regenerated.

- **Del**—Delete the remote access application.

Defining Remote Access Applications

User Permissions Needed
To manage, create, edit, and delete OAuth applications: “Manage Remote Access”

Before you start to use remote access applications, you need to consider the following:

- The version of the OAuth protocol to use.
- The type of application your developer is building. This determines which authentication flow to use.

For more information, see [Getting Started with Remote Access Applications](#) on page 316.

To define a remote access application:

1. To define a remote access application, click **Develop > Remote Access**, and click **New**.
2. Specify the name of the application. This is required. Salesforce.com recommends that the name of the remote access application match the name of the actual application.
3. Specify the **Callback URL**, which is required and represents the URL that the user will be returned to after they approve access for the application. This URL uses https or another protocol. This value can also be set to oob. It cannot use http, except if the callback is to your localhost.



Tip: OAuth 2.0 refers to the callback URL as `redirect_uri`.

4. If the application has a specific logo, you can specify that using the **Logo Image URL**. The URL must be secure (use https). The logo can be a maximum of 200x125 pixels. It is displayed on a white background.
5. Specify your **Contact Phone** and **Contact Email**. Contact Email is required.
6. In the **Info URL** field, you can specify a URL where users can go to get more information about the application. The URL must use https or http protocol, can't contain spaces, and has a maximum length of 2000 characters.
7. Enter a description of the application. When a user [grants access to an application](#), this description displays.
8. For applications, you can specify **No user approval required**. This means the application is automatically approved; that is, the end-user is never asked to approve access. This only works for end-users within your own organization.
9. If you're setting up an application to login and act on its own behalf, check the **Public Key Certificate** checkbox. You can then browse to locate and upload the authentication certificate issued by your identity provider.
10. Click **Save**.

When you save the remote access definition, the consumer key and consumer secret are automatically generated. The consumer key and consumer secret are available globally in all Database.com instances.



Note: After you save a remote access definition, it may take a few minutes before it becomes available.

The consumer should store the consumer key and consumer secret in their application. The keys are used in [authenticating a user using the remote access application](#).



Note: Even if you change the name of the application, the consumer key and consumer secret are **not** regenerated.

Managing Remote Access Application Requests

Remote Access Application Request

User Permissions Needed	
To manage, create, edit, and delete OAuth applications:	“Manage Remote Access”

The external application you are using is requesting access to your Database.com data. The external application has already been integrated into Database.com by your administrator.

To grant this application access to your Database.com data, click **Accept**.

If the description of the application does not match the application you are currently using or for any other reason you do not want to grant access to your data, click **Deny**.

If the currently logged in user is not you, click **Not you?** to log out the current user and log in as yourself.

You can only grant access to an external application a specific number of times. Generally, you grant access for every device you use, such as a laptop and a desktop computer. The default is five per application. If you've reached the limit for your organization, granting access to this application automatically revokes access to the token or tokens for this application that haven't been used for the longest period of time. The remote access application token or tokens that will be revoked display on the page.

After you have granted access to a remote access application, you can revoke it later by clicking **My Personal Information > Personal Information**, then in the Remote Access related section, click **Revoke**.

Remote Access Application Request Approved

User Permissions Needed	
To manage, create, edit, and delete OAuth applications:	“Manage Remote Access”

The external application you are using has requested access to your Database.com data, and you approved this request. Close the browser window and go back to the application you were using.

After you have granted access to a remote access application, you can revoke it later by clicking **My Personal Information > Personal Information**, then in the Remote Access related section, click **Revoke**.

Remote Access Application Request Denied

User Permissions Needed

To manage, create, edit, and delete OAuth applications: “Manage Remote Access”

The external application you are using has requested access to your Database.com data and you denied this access. You should log out of Database.com. You can go back to the originating application.

Managing OAuth

Authenticating Remote Access Application OAuth

User Permissions Needed

To manage, create, edit, and delete OAuth applications: “Manage Remote Access”

When a user requests their Database.com data from within the external application (the consumer’s page), the user must be authenticated by Database.com. There are several steps in each authentication flow, as dictated by the OAuth standard and what is trying to access Database.com.

Database.com supports the following authentication flows:

- [OAuth 1.0.A](#)—This version of OAuth has only one flow.
- [OAuth 2.0 Web server](#)—The Web server authentication flow is used by applications that are hosted on a secure server. A critical aspect of the Web server flow is that the server must be able to protect the consumer secret.
- [OAuth 2.0 user-agent](#)—The user-agent authentication flow is used by client applications (consumers) residing in the user’s device. This could be implemented in a browser using a scripting language such as JavaScript, or from a mobile device or a desktop application. These consumers cannot keep the client secret confidential.
- [OAuth 2.0 refresh token flow](#)—After the consumer has been authorized for access, they can use a refresh token to get a new access token (session ID.) This is only done after the consumer already has received an access token using either the Web server or user-agent flow.
- [OAuth 2.0 JWT Flow](#)

A JSON Web token (JWT) is a JSON-based security token encoding that enables identity and security information to be shared across security domains. Database.com supports JWT in requesting an OAuth 2.0 access token from a corporate portal or identity provider.

- [OAuth 2.0 SAML Bearer Flow](#)

The SAML bearer is a SAML Assertion based token. It is provided within the query string and is base 64 URL encoded. It includes the following sections:

- ◊ Subject/NameID that maps to the PRN / Username.
- ◊ Issuer element value that is mapped to the ISS as the client id
- ◊ Conditions where the NotOnOrAfter attribute is used to validate the expiration date of the assertion.
- ◊ Audience that is mapped to the aud and used for validation.
- [SAML assertion flow](#)—The SAML assertion flow is an alternative for organizations that are currently using SAML to access Database.com using single sign-on, and want to access the Web services API the same way. The SAML assertion flow can only be used inside a single organization. You do not have to create a remote access application to use this assertion flow.

- [OAuth 2.0 username and password](#)—The username-password authentication flow can be used as a replacement for an existing login when the consumer already has the user's credentials.



Caution: This OAuth authentication flow involves passing the user's credentials back and forth. Database.com recommends using this authentication flow only when necessary.

For all authentication flows, if a user is asked to authorize access and instead clicks the link indicating they are not the currently signed in user, the current user is logged out and the authorization flow restarts with authenticating the user.

OAuth 2.0 Endpoints

The two primary endpoints used with OAuth 2.0 are:

- Authorization—<https://login.database.com/services/oauth2/authorize>
- Token—<https://login.database.com/services/oauth2/token>

To revoke OAuth 2.0 tokens make a request to:

<https://login.database.com/services/oauth2/revoke>

See [Revoking OAuth Tokens](#) on page 327 for details on revoking access.

For [Test Database Overview](#), use `test.database.com` instead of `login.database.com`.

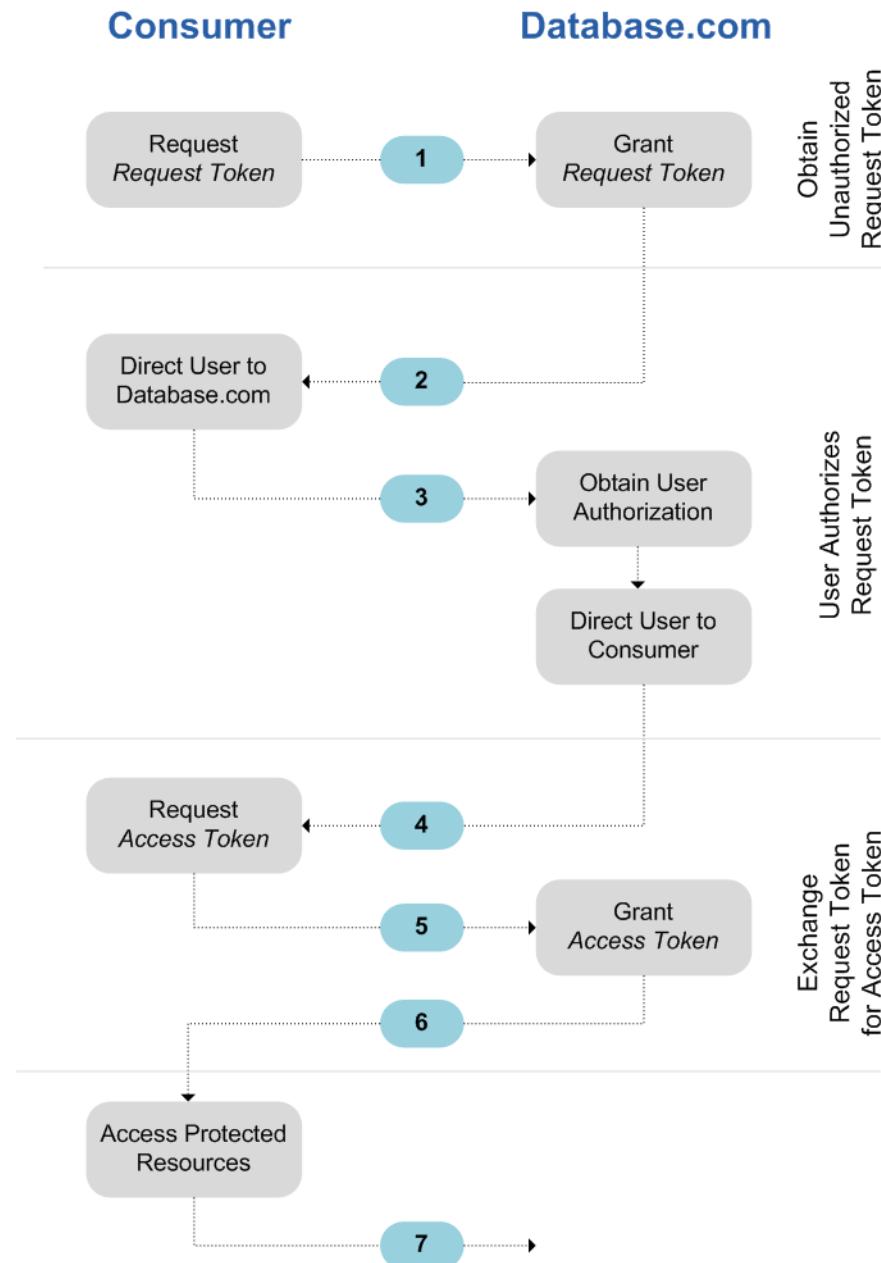
Managing OAuth 1.0

Managing OAuth 1.0.A Authentication Flow

OAuth 1.0.A Authentication Flow

User Permissions Needed
To manage, create, edit, and delete OAuth applications: “Manage Remote Access”

The following diagram displays the authentication flow steps for OAuth 1.0.A. The individual step descriptions follow. OAuth 1.0.A has a single authentication flow.



1. The consumer [requests a RequestToken](#). Database.com verifies the request and returns a request token.
2. The consumer should redirect the user to Database.com, where they are prompted to log in.
3. Database.com [authorizes the user](#).
4. Once the user is authorized, the consumer [requests an AccessToken](#).
5. Database.com verifies the request and grants the token.
6. After the token is granted, the consumer accesses the data either [through their application](#) or [through the Force.com Web services API](#).
7. Database.com verifies the request and allows access to the data.

The following sections go into more details about each of these steps.



Tip: To use a remote access application with a [Test Database Overview](#), use `test.database.com` instead of `login.database.com` in the following sections.

For the list of possible error codes returned by Database.com, see [OAuth 1.0.A Error Codes](#) on page 326.

Requesting a RequestToken

When a consumer makes an initial request to Database.com, a RequestToken is returned if the request is valid. The following steps contain more detail for the developer who is using a remote access application to request Database.com data.

1. A consumer application needs to access Database.com data and sends a request to `https://login.database.com/_nc_external/system/security/oauth/RequestTokenHandler`. The request contains the following:
 - A valid request for a RequestToken, which contains the following OAuth parameters.
 - ◊ `oauth_consumer_key`
 - ◊ `oauth_signature_method`—must be HMAC-SHA1.
 - ◊ `oauth_signature`
 - ◊ `oauth_timestamp`
 - ◊ `oauth_nonce`
 - ◊ `oauth_version`—optional, must be “1.0” if included
 - ◊ `oauth_callback`—must be one of the following:
 - URL hosted by the consumer, for example, `https://www.appirio.com/sfdc_accounts/access_token_ready.html`. Note that this URL uses https or another protocol. It cannot use http.
 - `oob`, meaning out of band.
 - A signature created according to the OAuth specification for HMAC-SHA1
2. After Database.com receives the request, Database.com:
 - Validates the request with its own copy of the consumer secret
 - Generates a response containing RequestToken and RequestTokenSecret in the HTTP body as name/value pairs
 - Sends the response back to the consumer

A RequestToken is only valid for 15 minutes, plus three minutes to allow for differences between machine clocks.
3. The consumer directs the user to a Database.com login page, as specified in the next section.

Authorizing the User

After the request from the consumer is made to Database.com, the user must be authenticated by Database.com before the process continues. The following contains more detailed steps about the login procedure for developers who are using a remote access application to request Database.com data.

1. The consumer redirects the user to the following location, where they are prompted to log in:
`https://login.database.com/setup/secur/RemoteAccessAuthorizationPage.apexp`. The appropriate GET query parameters are appended to this URL.
 - `oauth_token` – the RequestToken
 - `oauth_consumer_key`



Note: If an `oauth_callback` parameter is included, it is ignored.

2. The Remote Access Authorization page displays.
3. If the user approves access for the consumer, Database.com generates the AccessToken and AccessTokenSecret.



Note: The number of concurrent access tokens that can be granted by a user to an application is limited. The default is five per application per user. If this authorization exceeds the limit for the organization, the user is notified that their authorization automatically revokes the token or tokens for this application that haven't been used for the longest period of time.

4. Database.com verifies the callback URL (either specified in the remote access application definition pages or in the `oauth_callback` parameter from the previous stage). One of the following redirections occurs.
 - If the `oauth_callback` defined in the RequestToken is `oob` and the **Callback URL** field in the remote access application definition page has a valid value, the user is redirected to that URL.
 - If the `oauth_callback` defined in the RequestToken is a valid URL, the user is redirected to that URL.
5. The consumer is notified that the AccessToken and AccessTokenSecret are available either by receiving the verification token from Database.com or the validation code from the end user.

Requesting the AccessToken

Once the user has been authenticated, the consumer can exchange a RequestToken for an AccessToken. The following contains more detailed steps regarding the exchange of tokens for developers who are using a remote access application to request Database.com data.

1. The consumer makes an HTTPS GET or POST request to `https://login.database.com/_nc_external/system/security/oauth/AccessTokenHandler`, with the required parameters in the query or post data.
 - `oauth_consumer_key`
 - `oauth_signature_method`
 - `oauth_signature`
 - `oauth_timestamp`
 - `oauth_token`
 - `oauth_nonce`
 - `oauth_verifier`
 - `oauth_version`—optional, must be “1.0” if included
2. Database.com validates the following elements.
 - The consumer secret
 - The consumer key
 - The signature
 - That the RequestToken has never been used before
 - The timestamp (must be within 15 minutes, plus three minutes to allow for differences between machine clocks)
 - That the nonce has never used before
3. Upon validation, Database.com returns the AccessToken and AccessTokenSecret in the HTTP response body as name/value pairs.

Generating oauth_signature for Login

You can access Database.com using either the user interface, or using the API. The `oauth_signature` used for login is generated differently, depending on which method you use.

- User interface—use `https://login.database.com` for generating the signature
- API—use `https://login.database.com/services/OAuth/type/api-version` for generating the signature.
`type` must have one of the following values.
 - ◊ u—Partner WSDL
 - ◊ c—Enterprise WSDL

For example, `https://login.database.com/services/OAuth/u/17.0`.

Accessing Database.com Data Using the Consumer Application

Once the consumer possesses a valid AccessToken, a remote access application can request to access Database.com data. The following contains more detailed steps regarding accessing data for developers who are using a remote access application to request Database.com data.

1. The consumer makes an HTTPS POST request to `https://login.database.com`, with the required parameters in the authorization header.
 - `oauth_consumer_key`
 - `oauth_token`
 - `oauth_signature_method`
 - `oauth_signature`
 - `oauth_timestamp`
 - `oauth_nonce`
 - `oauth_version` (optional, must be “1.0” if included)
2. Database.com validates the request and sends a valid session ID to the consumer.

Accessing Database.com Data Using the API

Once the consumer possesses a valid AccessToken, a remote access application can request to access Database.com data using the Force.com Web services API.



Note: Your organization must have access to both the API and to the remote access application. Contact your salesforce.com representative for more information.

The following contains more detailed steps regarding accessing data for developers who are using a remote access application to request Database.com data.

1. The consumer makes an HTTPS POST request to Database.com.
 - The URL must have the following format:
`https://login.database.com/services/OAuth/type/api-version`.
`type` must have one of the following values.
 - ◊ u—Partner WSDL
 - ◊ c—Enterprise WSDL
`api-version` must be a valid API version.
 - The authorization header must have the following parameters.

- ◊ oauth_consumer_key
- ◊ oauth_token
- ◊ oauth_signature_method
- ◊ oauth_signature
- ◊ oauth_timestamp
- ◊ oauth_nonce
- ◊ oauth_version (optional, must be “1.0” if included)

2. Database.com validates the request and sends a valid session ID to the consumer. The response header includes the following.

```
<response>
<metadataServerUrl>https://nal-api.salesforce.com/services/Soap/m/17.0/00D300000006qjK
</metadataServerUrl>
<sandbox>false</sandbox>
<serverUrl>https://nal-api.salesforce.com/services/Soap/u/17.0/00D300000006qjK
</serverUrl>
<sessionId>00D300000006qrN!AQoAQJTMzwTa67tGgQck1ng_xgMSuWVBpFwZ1xUq2kLjMYg6Zq
    GTS8Ezu_C3w0pdT1DMyHiJgB6fbhhEPxKjGqlYnlROIUs1</sessionId>
</response>
```

OAuth 1.0.A Error Codes

Database.com returns the following error codes during the [OAuth 1.0.A Authentication Flow](#). The returned error code is based on the error received.

Fault Code	Error	Notes
1701	Failed: Nonce Replay Detected	A Nonce can only be used once.
1702	Failed: Missing Consumer Key Parameter	
1703	Failed: Invalid Access Token	
1704	Failed: Version Not Supported	You must specify 1.0 for the <code>oauth_version</code> parameter.
1705	Failed: Invalid Timestamp	The timestamp is one of the following: missing, in the future, too old, or malformed.
1706	Failed: Invalid Nonce	The Nonce is missing.
1707	Failed: Missing OAuth Token Parameter	
1708	Failed: IP Address Not Allowed	
1709	Failed: Invalid Signature Method	The RequestToken contains an invalid <code>oauth_signature_method</code> parameter.
1710	Failed: Invalid Callback URL	The RequestToken contains an invalid <code>oauth_callback</code> parameter. Value must be either <code>oob</code> or a valid URL that uses https.
1711	Failed: Invalid Verifier	The AccessToken. contains an invalid <code>oauth_verifier</code> parameter.

Fault Code	Error	Notes
1712	Failed: Get Access Token Limit Exceeded	Can only attempt to exchange a RequestToken for an AccessToken three times.
1713	Failed: Consumer Deleted	The remote access application has been deleted from the Database.com organization.
1716	Failed: OAuth Api Access Disabled	Either the Force.com Web services API is not enabled for the organization, or OAuth API access has been disabled for the organization.

Managing OAuth 2.0

Revoking OAuth Tokens

When users request their data from within the external application (the consumer's page), they are authenticated. You can revoke their access tokens, or the refresh token and all related access tokens. Developers may use this feature when configuring a Log Out button in their application.

Revoking Tokens

To revoke OAuth 2.0 tokens, use the relocation endpoint:

```
https://login.database.com/services/oauth2/revoke
```

Construct a POST request that includes the following parameters using the `application/x-www-form-urlencoded` format in the HTTP request entity-body. For example:

```
POST /revoke HTTP/1.1
Host: https://login.database.com/services/oauth2/revoke
Content-Type: application/x-www-form-urlencoded

token=currnettoken
```

If an access token is included, we invalidate it and revoke the token. If a refresh token is included, we revoke it as well as any associated access tokens.

The authorization server indicates successful processing of the request by returning an HTTP status code 200. For all error conditions, a status code 400 is used along with one of the following error responses.

- `unsupported_token_type`—token type not supported
- `invalid_token`—client identifier invalid

For [Test Database Overview](#), use `test.database.com` instead of `login.database.com`.

JSONP Support

The revocation endpoint also accepts GET requests with an additional `callback` parameter, and returns the response with content type `application/javascript`. For example:

```
https://login.database.com/services/oauth2/revoke?token=XXXXX&callback=package.myCallback
```

If the request is successful, a callback is sent to the JavaScript function set in the `callback` parameter of the GET:

```
package.myCallback({});
```

If the response is not successful, a callback is sent with an error code:

```
package.myCallback({"error":"invalid_token"});
```

Blind GETs

We also support blind GET requests with the query string parameter `token` and the current token. If an access token is included, we invalidate it and revoke the token. If a refresh token is included, we revoke it as well as any associated access tokens. For example:

```
https://login.database.com/services/oauth2/revoke?token=currrenttokenID
```

The authorization server indicates successful processing of the request by returning an HTTP status code 200. For all error conditions, status code 400 is used.

Managing OAuth 2.0 Web Server Authentication Flow

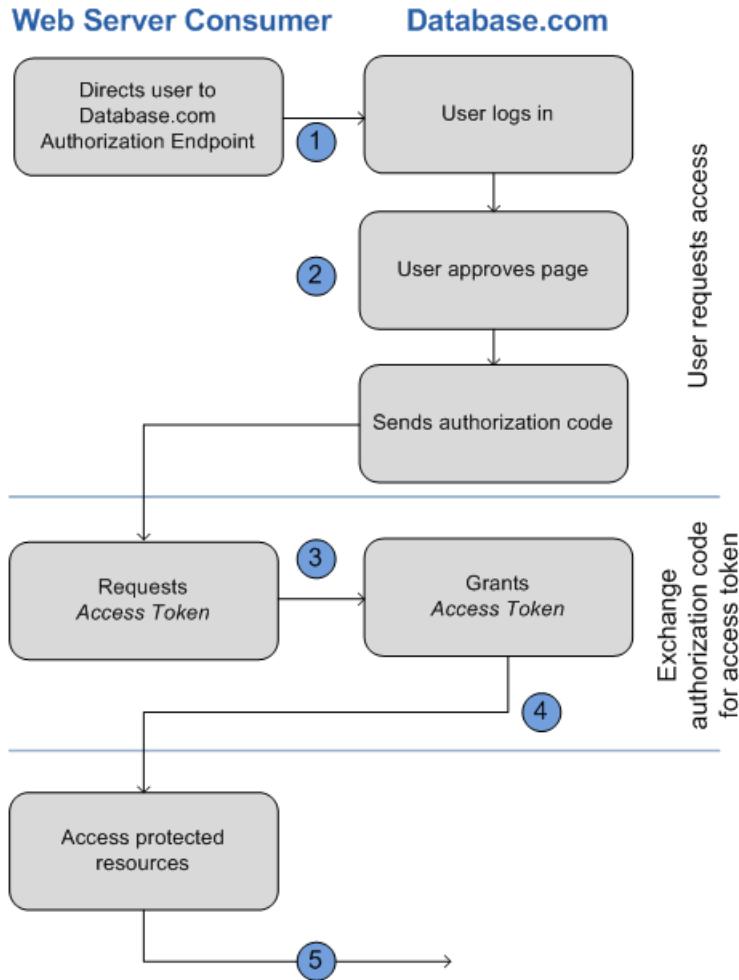
OAuth 2.0 Web Server Authentication Flow

User Permissions Needed

To manage, create, edit, and delete OAuth applications:	“Manage Remote Access”
---	------------------------

The Web server authentication flow is used by applications that are hosted on a secure server. A critical aspect of the Web server flow is that the server must be able to protect the consumer secret.

The following diagram displays the authentication flow steps for Web server clients. The individual step descriptions follow.



1. The Web server redirects the user to Database.com to authenticate and authorize the server to [access data on their behalf](#).
2. After the user approves access, the Web server [receives the callback](#).
3. After obtaining the authorization code, the consumer passes back the authorization code to [obtain an access token](#).
4. After validating the authorization code, Database.com [passes back a response token](#). If there was no error, the response token includes an access code, a refresh token, and additional information.
5. After the token is granted, the consumer accesses their data.

After a consumer has an access token, they can [use the access token](#) to access Database.com data on the end user's behalf and use a [refresh token](#) to get a new access token if it becomes invalid for any reason.

Redirect User to Obtain Access Authorization

To obtain authorization from the user to access Database.com data on his or her behalf, the client redirects the user's browser to the authorization endpoint with the following parameters:

- `response_type`—Value must be `code` for this flow.
- `client_id`—Consumer key from the remote access application definition.
- `scope`—The scope parameter allows you to fine-tune what the client application can access in a Database.com organization. See [Scope Parameter Values](#) on page 335 for valid parameters.
- `redirect_uri`—URI to redirect the user to after approval. This must match the value in the `Callback URL` field in the remote access application definition exactly, or approval fails. This value must be URL encoded.

- **state**—Any state the consumer wants reflected back to it after approval, during the callback. This parameter is optional. This value must be URL encoded.
- **immediate**—Determines whether the user should be prompted for login and approval. This parameter is optional. The value must be `true` or `false` if specified. Default value is `false`. Note the following:
 - ◊ If set to `true`, and if the user is currently logged in and has previously approved the `client_id`, Database.com skips the approval step.
 - ◊ If set to `true` and the user is not logged in or has not previously approved the client, Database.com immediately terminates with the `immediate_unsuccessful` error code.
- **display**—Changes the login and authorization pages' display type. This parameter is optional. The only values Database.com supports are:
 - ◊ `page`—Full-page authorization screen. This is the default value if none is specified.
 - ◊ `popup`—Compact dialog optimized for modern web browser popup windows.
 - ◊ `touch`—mobile-optimized dialog designed for modern smartphones such as Android and iPhone.
 - ◊ `mobile`—mobile optimized dialog designed for less capable smartphones such as BlackBerry OS 5.

In order to initiate the flow, the Web server generally forms a link, or sends an HTTP redirect to the browser. The following is an example of a request to an authorization endpoint from a Web server client:

```
https://login.database.com/services/oauth2/authorize?response_type=code&client_id=3MVG91KcPoNINVBiPjdw1J9LLM82HnFVVX19KY1uA5mu0QqEWhqKpoW3svG3XHrXDjCQjK1mdgAvhCscA9GE&redirect_uri=https%3A%2F%2Fwww.mysite.com%2Fcode_callback.jsp&state=mystate
```

If the user is logged in, Database.com redirects them to the approval page. If the user is not logged in, they are asked to log in, then redirected to the approval page where they grant access to the application. If the user has already approved access once, they don't have to approve access again.

Web Server Received Callback

Once the user approves the access, they are redirected to the URI specified in `redirect_uri` with the following values in the query string:

- `code`—Authorization code the consumer must use to obtain the access and refresh tokens
- `state`—State that was passed into the approval step. This isn't included if the `state` parameter wasn't included in the original query string.

If the user has already approved the access once, they do not have to approve access again.

The following is an example of the request received by the `redirect_uri`:

```
https://www.mysite.com/code_callback.jsp?code=aPrxsmIEeqM9&state=mystate
```

If the user denies the application, they are redirected to the `redirect_uri` with the following values in the query string:

- `error`—Value is `access-denied`.
- `state`—State that was passed into the approval step. This isn't included if the `state` parameter wasn't included in the original query string.

For example:

```
https://www.mysite.com/code_callback.jsp?error=access-denied&state=mystate
```

If the user denies access, or an error occurs during this step, the response contains an error message containing these parts:

- `error`—Error code
- `error_description`—Description of the error with additional information.
 - ◊ `unsupported_response_type`—response type not supported
 - ◊ `invalid_client_id`—client identifier invalid
 - ◊ `invalid_request`—HTTPS required
 - ◊ `invalid_request`—must use HTTP GET
 - ◊ `access_denied`—end-user denied authorization
 - ◊ `redirect_uri_missing`—redirect_uri not provided
 - ◊ `redirect_uri_mismatch`—redirect_uri mismatch with remote access application definition
 - ◊ `immediate_unsuccessful`—immediate unsuccessful
 - ◊ `invalid_scope`—requested scope is invalid, unknown, or malformed
- `state`—State that was passed into the approval step. This isn't included if the `state` parameter wasn't included in the original query string.

Web Server Exchanges Verification Code for Access Token

After obtaining the authorization code, the Web server exchanges the authorization code for an access token.

The consumer should make a POST directly to the token endpoint, with the following parameters:

- `grant_type`—Value must be `authorization_code` for this flow.
- `client_id`—Consumer key from the remote access application definition.
- `client_secret`—Consumer secret from the remote access application definition.
- `redirect_uri`—URI to redirect the user to after approval. This must match the value in the `Callback URL` field in the remote access application definition exactly, and is the same value sent by the initial redirect. See [Redirect User to Obtain Access Authorization](#) on page 329.
- `code`—Authorization code obtained from the callback after approval.
- `format`—Expected return format. This parameter is optional. The default is `json`. Values are:
 - ◊ `urlencoded`
 - ◊ `json`
 - ◊ `xml`

The following is an example of the POST body sent out-of-band:

```
POST /services/oauth2/token HTTP/1.1
Host: login.database.com
grant_type=authorization_code&code=aPrxsmIEeqM9PiQroGEWx1UiMQd95_5JUZ
VEhsOFhS8EVvbfYBBJli2W5fn3zbo.8hojaNW_1g%3D%3D&client_id=3MVG91KcPoNI
NVBIPJjdwlJ9LLM82HnFVX19KY1uA5mu0QqEWhqKpoW3svG3XHrXDicQjK1mdgAvhCs
cA9GE&client_secret=1955279925675241571&
redirect_uri=https%3A%2F%2Fwww.mysite.com%2Fcode_callback.jsp
```

Instead of using the `format` parameter, the client can also specify the returned format in an `accept-request` header using one of the following:

- `Accept: application/json`
- `Accept: application/xml`
- `Accept: application/x-www-form-urlencoded`

Note the following:

- Wildcard accept headers are allowed. `*/*` is accepted and returns JSON.

- A list of values is also accepted and is checked left-to-right. For example: `application/xml, application/json, application/html, */*` returns XML.
- The `format` parameter takes precedence over the `accept` request header.

Database.com Responds with Access Token

After the request is verified, Database.com sends a response to the client. The following parameters are in the body of the response:

- `access_token`—Database.com session ID that can be used with the Web services API.
- `refresh_token`—Token that can be used in the future to [obtain new access tokens](#) (sessions). **This value is a secret. You should treat it like the user's password and use appropriate measures to protect it.**
- `instance_url`—URL indicating the instance of the user's organization. In this example, the instance is na1: `https://na1.salesforce.com`.
- `id`—Identity URL that can be used to both identify the user as well as query for more information about the user. See [Using Identity URLs](#) on page 282.
- `signature`—Base64-encoded HMAC-SHA256 signature signed with the consumer's private key containing the concatenated ID and `issued_at`. This can be used to verify the identity URL was not modified since it was sent by the server.
- `issued_at`—When the signature was created.

The following is an example response from Database.com:

```
{"id":"https://login.database.com/id/00Dx000000BV7z/005x0000012Q9P",
"issued_at":"1278448101416","refresh_token":"5Aep8614iLM.Dq661ePDmPEgaAW9
Oh_L3JKkDpB4xReb54_pZebnUG0h6Sb4KUVdpNtWEofWM39yg==","instance_url":
"https://na1.salesforce.com","signature":"CMJ4l+CCaPQiKjoOEwEig9H4wqhpulSk
4J2urAe+fVg=","access_token":"00Dx000000BV7z!AR8AQP0jITN80ESEs5EbazTFG0R
NBaTlcWk7TrqoDjoNIWQ2ME_stZzBjfmoE6zMh6y8PIW4eWze9JksNEkWU1.Cju7m4"}
```

If an error occurs during this step, the response contains an error message with these parts:

- `error`—Error code
- `error_description`—Description of the error with additional information.
 - ◊ `unsupported_response_type`—response type not supported
 - ◊ `invalid_client_id`—client identifier invalid
 - ◊ `invalid_request`—HTTPS required
 - ◊ `invalid_request`—must use HTTP POST
 - ◊ `invalid_client_credentials`—client secret invalid
 - ◊ `invalid_grant`—invalid authorization code
 - ◊ `invalid_grant`—IP restricted or invalid login hours
 - ◊ `redirect_uri_mismatch`—`redirect_uri` not provided
 - ◊ `redirect_uri_mismatch`—`redirect_uri` mismatch with remote access application definition
 - ◊ `inactive_user`—user has been set to inactive by the administrator
 - ◊ `inactive_org`—organization is locked, closed, or suspended
 - ◊ `rate_limit_exceeded`—number of login attempts has been exceeded

Any login error not listed receives a generic authentication failure with text describing the error. For example, `LOGIN_ERROR_INVALID_PASSWORD` would have the following error response:

```
{"error":"authentication_failure","error_description":"invalid password"}
```

Managing OAuth 2.0 User-Agent Flow

OAuth 2.0 User-Agent Flow

User Permissions Needed	
To manage, create, edit, and delete OAuth applications:	“Manage Remote Access”

The user-agent authentication flow is used by client applications (consumers) residing in the user's device. This could be implemented in a browser using a scripting language such as JavaScript, or from a mobile device or a desktop application. These consumers cannot keep the client secret confidential. The authentication of the consumer is based on the user-agent's same-origin policy.

Unlike the other authentication flows, the client application receives the access token in the form of an HTTP redirection. The client application requests the authorization server to redirect the user-agent to another web server or local resource accessible to the user-agent, which is capable of extracting the access token from the response and passing it to the client application. Note that the token response is provided as a hash (#) fragment on the URL. This is for security, and prevents the token from being passed to the server, as well as to other servers in referral headers.

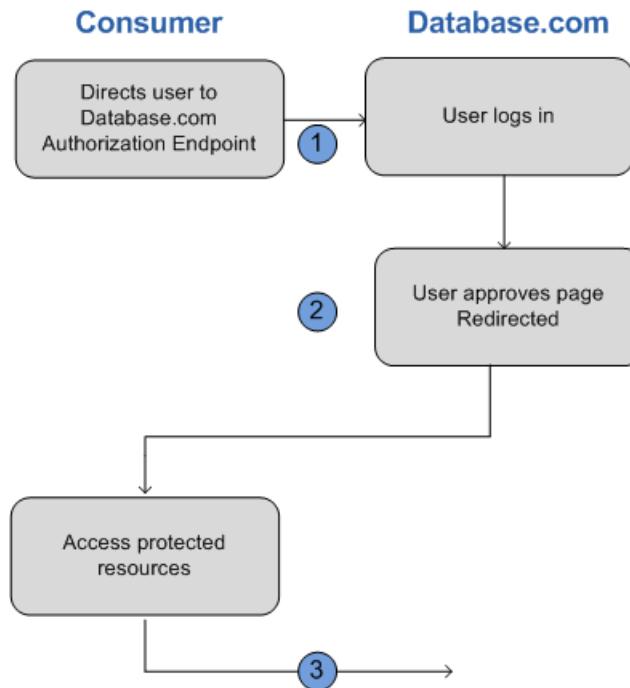
This user-agent authentication flow doesn't utilize the client secret since the client executables reside on the end-user's computer or device, which makes the client secret accessible and exploitable.



Caution: Because the access token is encoded into the redirection URI, it might be exposed to the end-user and other applications residing on the computer or device.

If you are authenticating using JavaScript, Database.com recommends calling `window.location.replace()` to remove the callback from the browser's history.

The following diagram displays the authentication flow steps for Web server clients. The individual step descriptions follow.



1. The client application directs the user to Database.com to authenticate and authorize the application.

- The user must always approve access for this authentication flow. After approving access, the application receives the callback from Database.com.

After a consumer has an access token, they can [use the access token](#) to access Database.com data on the end user's behalf and a [refresh token](#) to get a new access token if it becomes invalid for any reason.

The user-agent flow does not support out-of-band posts.

Direct User to Database.com to Obtain Access Token

To obtain authorization from the user to access Database.com data on his or her behalf, the client directs the user to the authorization endpoint with the following parameters:

- `response_type`—Value must be `token` for this flow.
- `client_id`—Consumer key from the remote access application definition.
- `redirect_uri`—URI to redirect the user to after approval. This must match the value in the `Callback URL` field in the remote access application definition exactly. This value must be URL encoded.
- `state`—Any state the consumer wants reflected back to it after approval, during the callback. This parameter is optional.
- `scope`—The scope parameter allows you to fine-tune what the client application can access in a Database.com organization. See [Scope Parameter Values](#) on page 335 for valid parameters.
- `display`—Changes the login page's display type. This parameter is optional. The only values Database.com supports are:
 - `page`—Full-page authorization screen. This is the default value if none is specified.
 - `popup`—Compact dialog optimized for modern web browser popup windows.
 - `touch`—mobile-optimized dialog designed for modern smartphones, such as Android and iPhone.

The following is an example URL where the user is directed to:

```
https://login.database.com/services/oauth2/authorize?response_type=token&
client_id=3MVG9lKcPoINVBIJjdw1J9LLJbP_pgwoJYyuisjQhr_LLurNDv7AgQvDTZwCoZuD
ZrXcPCmBv4o.8ds.5iE&redirect_uri=https%3A%2F%2Fwww.mysite.com%2Fuser_callback.jsp&
state=mystate
```

User Approves Access and Client Receives Callback from Database.com

The user is asked to log in to Database.com if they are not already logged in. Then Database.com displays an approval page, asking the user to approve the application access. If the user approves the access, they are redirected to the URI specified in `redirect_uri` with the following values after the hash sign (#). This is not a query string.

- `access_token`—Database.com session ID that can be used with the Web services API.
- `refresh_token`—Token that can be used in the future to [obtain new access tokens](#) (sessions). **This value is a secret. You should treat it like the user's password and use appropriate measures to protect it.**



Note: The refresh token for the user-agent flow is only issued under one of the following circumstances:

- The redirect URL uses a custom protocol that is not HTTPS.
- The redirect URL is exactly `https://login.database.com/services/oauth2/success`, or on Test Database Overview, `https://test.database.com/services/oauth2/success`.

- `instance_url`—URL indicating the instance of the user's organization. In this example, the instance is na1: `https://na1.salesforce.com`.
- `id`—Identity URL that can be used to both identify the user as well as query for more information about the user. See [Using Identity URLs](#) on page 282.

- **signature**—Base64-encoded HMAC-SHA256 signature signed with the consumer's private key containing the concatenated ID and `issued_at`. This can be used to verify the identity URL was not modified since it was sent by the server.
- **issued_at**—When the signature was created.

The following is an example of the callback from the server. Note the response is behind a hash, rather than as HTTP query parameters:

```
https://www.mysite.com/user_callback.jsp#access_token=00Dx0000000BV7z%21AR8
AQBM8J_xr9kLqmZIRyQxZgLcM4HVi41aGtW0qW3JCzf5xdTGGGSoVim8FFJkZEqxbjaFbberKGk
8v8AnYrvChG4qJbQo8&refresh_token=5Aep8614iLM.Dq661ePDmPEgaAW9Oh_L3JKkDpB4xR
eb54_pZfVti1dPEk8aimw4Hr9ne7VXXVSIQ%3D%3D&expires_in=7200&state=mystate
```

If the user denies access or an error occurs during this step, they are redirected to the `redirect_uri` with an error code and the description of the error in the URI, after the hash tag (#). This is not a query string.

- **error**—Error code
- **error_description**—Description of the error with additional information.
 - ◊ `unsupported_response_type`—response type not supported
 - ◊ `invalid_client_id`—client identifier invalid
 - ◊ `invalid_request`—HTTPS required
 - ◊ `invalid_request`—must use HTTP GET
 - ◊ `invalid_request`—out-of-band not supported
 - ◊ `access_denied`—end-user denied authorization
 - ◊ `redirect_uri_missing`—`redirect_uri` not provided
 - ◊ `redirect_uri_mismatch`—`redirect_uri` mismatch with remote access object
 - ◊ `immediate_unsuccessful`—immediate unsuccessful
 - ◊ `invalid_grant`—invalid user credentials
 - ◊ `invalid_grant`—IP restricted or invalid login hours
 - ◊ `inactive_user`—user is inactive
 - ◊ `inactive_org`—organization is locked, closed, or suspended
 - ◊ `rate_limit_exceeded`—number of logins exceeded
 - ◊ `invalid_scope`—requested scope is invalid, unknown, or malformed

Any login error not listed receives a generic authentication failure with text describing the error. For example, `LOGIN_ERROR_INVALID_PASSWORD` would have the following error response:

```
{"error": "authentication_failure", "error_description": "invalid password"}
```

- **state**—State that was passed into the approval step. This isn't included if the `state` parameter wasn't included in the original query string.

The following is an example error redirect URI:

```
https://www.mysite.com/user_callback.jsp#error=access_denied&state=mystate
```

Scope Parameter Values

User Permissions Needed

To manage, create, edit, and delete OAuth applications: “Manage Remote Access”

The scope parameter allows you to fine-tune what the client application can access in a Database.com organization. You can specify more than one scope at a time. Separate them with a space. If you do not provide a scope, you are implicitly setting the scope parameter to include all of the following values:

- id
- api
- refresh_token (only applicable if a refresh token would normally be issued)

Regardless of which scope you pass, you always have access to the identity URLs.

Here is a sample request setting the scope parameter with the api, id, and web values:

```
http://localhost:9088/services/oauth2/authorize?response_type=token&client_id=3MVG9lKcPonINVBKv6EgVJiF.snSDwh6_2wSS7BrOhHGEJkC_&redirect_uri=http://localhost:9088/qa/security/oauth/useragent_flow_callback.jsp&scope=api id web
```

The valid scope values are:

Value	Description
api	Allows access to the current, logged-in user's account over the APIs, such as the REST API or Bulk API.
chatter_api	Allows access to only the Chatter API URLs.
full	Allows access to all data accessible by the current, logged-in user.
id	Allows access only to the identity URL service. See Using Identity URLs on page 282.
refresh_token	Allows a refresh token to be returned if you are eligible to receive one.
web	Allows the ability to use the access_token on the Web.

Managing OAuth 2.0 Refresh Token Flow

OAuth 2.0 Refresh Token Flow

User Permissions Needed

To manage, create, edit, and delete OAuth applications: “Manage Remote Access”

After the consumer has been authorized for access, they can use a refresh token to get a new access token (session ID.) This is only done after the consumer already has received an access token using either the Web server or user-agent flow. It is up to the consumer to determine when an access token is no longer valid, and when to apply for a new one.

The following are the steps for the refresh token authentication flow. More detail about each step follows:

1. The consumer uses the existing refresh token to [request a new access token](#).
2. After the request is verified, Database.com [sends a response](#) to the client.

Consumer Requests Updated Access Token

A consumer can use the refresh token to get a new session as needed.

The consumer should make POST request to the token endpoint, with the following parameters:

- `grant_type`—Value must be `refresh_token` for this flow.
- `refresh_token`—Refresh token from the approval step.
- `client_id`—Consumer key from the remote access application definition.
- `client_secret`—Consumer secret from the remote access application definition. This parameter is optional.
- `format`—Expected return format. This parameter is optional. The default is `json`. Values are:

- ◊ `urlencoded`
- ◊ `json`
- ◊ `xml`

The following example is the out-of-band POST body to the token endpoint:

```
POST /services/oauth2/token HTTP/1.1
Host: https://login.database.com/
grant_type=refresh_token&client_id=3MVG91KcPoNINVBJJdw1J9LLM82HnFVVX19KY1uA5mu0
QqEWhqKpoW3svG3XHrXDiCQjK1mdgAvhCscA9GE&client_secret=1955279925675241571
&refresh_token=your token here
```

Instead of using the `format` parameter, the client can also specify the returned format in an accept-request header using one of the following:

- `Accept: application/json`
- `Accept: application/xml`
- `Accept: application/x-www-form-urlencoded`

Database.com Server Sends a Response

After the request is verified, Database.com sends a response to the client. The following parameters are in the body of the response:

- `access_token`—Database.com session ID that can be used with the Web services API.
- `instance_url`—URL indicating the instance of the user's organization. In this example, the instance is `na1: https://na1.salesforce.com`.
- `id`—Identity URL that can be used to both identify the user as well as query for more information about the user. See [Using Identity URLs](#) on page 282.
- `signature`—Base64-encoded HMAC-SHA256 signature signed with the consumer's private key containing the concatenated ID and `issued_at`. This can be used to verify the identity URL was not modified since it was sent by the server.
- `issued_at`—When the signature was created.

The following is a JSON example response from Database.com:

```
{ "id": "https://login.database.com/id/00Dx0000000BV7z/005x00000012Q9P",
"issued_at": "1278448384422", "instance_url": "https://na1.salesforce.com",
"signature": "SSSbLO/gBhmmNUvN180DBDFYHzakxOMggYtuhDPsc=",
"access_token": "00Dx0000000BV7z!AR8AQPOjITN80ESEsj5EbaZTFG0RNBaT1cyWk7T
rqoDjoNIWQ2ME_sTZZBjfmOE6zMh6y8PIW4eWze9JksNEkWU1.Cju7m4" }
```

The following is an XML example response:

```
<oauth>
  <access_token>00Dx0000000BV7z!AR8AQPOjITN80ESEsj5EbaZTFG0RNB
    aT1cyWk7TrqoDjoNIWQ2ME_stZzBjfmOE6zMh6y8PIW4eWze9JksNEkWU1.Cju7m4
  </access_token>
  <instance_url>https://na1.salesforce.com</instance_url>
  <id>https://login.database.com/id/00Dx0000000BV7z/005x00000012Q9P</id>
  <issued_at>1278448101416</issued_at>
  <signature>CMJ4l+CCaPQiKjoOEwEig9H4wqhpulSk4J2urAe+fVg=</signature>
</oauth>
```

The following is an URL encoded example:

```
access_token=00Dx0000000BV7z!AR8AQPOjITN80ESEsj5EbaZTFG0
  RNBaT1cyWk7TrqoDjoNIWQ2ME_stZzBjfmOE6zMh6y8PIW4eWze
  9JksNEkWU1.Cju7m4
& instance_url=https%3A%2F%2Fn1.salesforce.com
& id=https%3A%2F%2Flogin.database.com%2Fid%2F00Dx0000000BV
  7z%2F005x00000012Q9P
& issued_at=1278448101416
& signature=CMJ4l%2BCCaPQiKjoOEwEig9H4wqhpulSk4J2urAe%2BfVg%3D
```

If a problem occurs during this step, the response contains an error message with these parts:

- `error`—Error code
- `error_description`—Description of the error with additional information.
 - ◊ `unsupported_response_type`—response type not supported
 - ◊ `invalid_client_id`—client identifier invalid
 - ◊ `invalid_request`—HTTPS required
 - ◊ `invalid_request`—must use HTTP POST
 - ◊ `invalid_client_credentials`—client secret invalid
 - ◊ `invalid_request`—secret type not supported
 - ◊ `invalid_grant`—expired access/refresh token
 - ◊ `invalid_grant`—IP restricted or invalid login hours
 - ◊ `inactive_user`—user is inactive
 - ◊ `inactive_org`—organization is locked, closed, or suspended
 - ◊ `rate_limit_exceeded`—number of logins exceeded
 - ◊ `invalid_scope`—requested scope is invalid, unknown, or malformed

Any login error not listed receives a generic authentication failure with text describing the error. For example, `LOGIN_ERROR_INVALID_PASSWORD` would have the following error response:

```
{"error": "authentication_failure", "error_description": "invalid password"}
```

The following is an example of an error response:

```
{"error": "invalid_client_credentials", "error_description": "client secret invalid"}
```

Managing OAuth 2.0 Username-Password Flow

OAuth 2.0 Username-Password Flow

User Permissions Needed

To manage, create, edit, and delete OAuth applications: “Manage Remote Access”

The username-password authentication flow can be used as a replacement for an existing login when the consumer already has the user's credentials.

 **Caution:** This OAuth authentication flow involves passing the user's credentials back and forth. Database.com recommends using this authentication flow only when necessary.

The following are the steps for the username-password authentication flow. More detail about each step follows:

1. The consumer uses the end-user's username and password to [request an access token](#) (session ID.)
2. After the request is verified, Database.com [sends a response](#) to the client.

After a consumer has an access token, they can [use the access token](#) to access Database.com data on the end-user's behalf.

Request an Access Token

The consumer can use the end-user's username and password to request an access token, which can be used as a session ID.

The consumer should make an out-of-band POST request to the token endpoint, with the following parameters:

- `grant_type`—Value must be `password` for this flow.
- `client_id`—Consumer key from the remote access application definition.
- `client_secret`—Consumer secret from the remote access application definition.
- `username`—End-user username.
- `password`—End-user password



Note: When using the username-password flow with the API, be sure to create a field in the username and password login screen where users can input their security token. You must concatenate their password and token when passing the request for authentication. For more information about security tokens, see [Resetting Your Security Token](#) on page 79

- `format`—Expected return format. This parameter is optional. The default is `json`. Values are:
 - ◊ `urlencoded`
 - ◊ `json`
 - ◊ `xml`

The following is an example of the body of the out-of-band POST:

```
grant_type=basic-credentials&client_id=3MVG91KcPoNINVBiPJjdw1J9LLM82Hn
FVVX19KY1uA5mu0QqEWhqKpoW3svG3XHrXDjCQjK1mdgAvhCscA9GE&client_secret=
1955279925675241571&username=testuser%40database.com&password=mypassword
```

Send Response

After the request is verified, Database.com sends a response to the client. The following parameters are in the body of the response:

- `access_token`—Database.com session ID that can be used with the Web services API.
- `instance_url`—URL indicating the instance of the user's organization. In this example, the instance is `na1: https://na1.salesforce.com`.

- `id`—Identity URL that can be used to both identify the user as well as query for more information about the user. See [Using Identity URLs](#) on page 282.
- `signature`—Base64-encoded HMAC-SHA256 signature signed with the consumer's private key containing the concatenated ID and `issued_at`. This can be used to verify the identity URL was not modified since it was sent by the server.
- `issued_at`—When the signature was created.

Note: No refresh token is sent with this response.



The following is an example response:

```
{"id":"https://login.database.com/id/00Dx000000BV7z/005x0000012Q9P",
"issued_at":"1278448832702","instance_url":"https://na1.salesforce.com",
"signature":"0CmxinZir53Yex7nE0TD+zMpvIWYGb/bdJh6XfOH6EQ=","access_token":
"00Dx000000BV7z!AR8AQAxo9UfVkh8A1V0Gomt9Czx9LjHnSSpwBMmbRcgKFmx0tvxjTrKW1
9ye6PE3Ds1eQz3z8jr3W7_VbWmEu4Q8TVGSTHxs"}
```

If a problem occurs during this step, the response contains an error message with these parts:

- `error`—Error code
- `error_description`—Description of the error with additional information.
 - ◊ `unsupported_response_type`—response type not supported
 - ◊ `invalid_client_id`—client identifier invalid
 - ◊ `invalid_request`—HTTPS required
 - ◊ `invalid_request`—must use HTTP POST
 - ◊ `invalid_client_credentials`—client secret invalid
 - ◊ `invalid_grant`—invalid user credentials
 - ◊ `invalid_grant`—IP restricted or invalid login hours
 - ◊ `inactive_user`—user is inactive
 - ◊ `inactive_org`—organization is locked, closed, or suspended
 - ◊ `rate_limit_exceeded`—number of logins exceeded
 - ◊ `invalid_scope`—requested scope is invalid, unknown, or malformed

Any login error not listed receives a generic authentication failure with text describing the error. For example, `LOGIN_ERROR_INVALID_PASSWORD` would have the following error response:

```
{"error":"authentication_failure","error_description":"invalid password"}
```

The following is an example of a returned error:

```
{"error":"invalid_client_credentials","error_description":"client secret invalid"}
```

Managing OAuth 2.0 JWT Flow

OAuth 2.0 JWT Flow

User Permissions Needed

To manage, create, edit, and delete OAuth applications: “Manage Remote Access”

A JSON Web token (JWT) is a JSON-based security token encoding that enables identity and security information to be shared across security domains. Database.com supports JWT in requesting an OAuth 2.0 access token from a corporate portal or identity provider.

The JWT token has three token segments:

- Header segment (base64url encoded JSON) — defines algorithm used to encode the signature and the type of token.
- Body (base64url encoded JSON) — client id (iss), username(prn) and expiration times of the token.
- Signature (base64url encoded JSON) — Encrypted string of Header + '.' + body in their encoded forms.

The general flow for a JWT bearer token is similar to a refresh token flow within OAuth. It is a way to exchange authentication information after a prior client is authorized by the user. An exchange is done from one SSO provider to another using shared secrets that don't go across the user's browser.

The following are the steps for the JWT token authentication flow.

1. Client creates Remote Access application and signs with shared secret.
2. Client sends over the bearer token through a direct call to token servlet.
3. Token servlet validates signature with the shared secret.
4. Access token is created and sent back to client.

Client Creates Remote Access Application

During this step, the client must create a Remote Access application in an organization so they can get a consumer secret. Using this secret, the client writes an application that generates a JWT token with one of the following:

- Private key for RSA-256 based signature — The public key is uploaded using Remote Access.
- Consumer secret for HMAC-256 signature.

Client Sends JWT Token

The client creates and sends the JWT request to Database.com with the following parameters:

- Prn (principal) claim — Username of the user getting the token. If `client_id`, then client is authenticating as itself. If `user`, then it's a regular impersonation flow with `username`, `userid/orgid`, `userurl`.
- Iss (issuer) claim — Consumer key from the remote access.
- Exp (expirations) claim — When the token expires. this limits the time window during which the JWT can be used.
- Aud (audience) claim — contains a URI reference that identifies the authorization server as the intended audience. The authorization server MUST verify that it is an intended audience for the JWT.
- Iat (issued at, optional) claim — the UTC time at which the JWT was issued.

The parameter namespace for `grant_type` is `urn:ietf:params:oauth:grant-type:jwt-bearer`.



Note: Scopes are not supported in this flow.

Here is a sample JWT request:

```
{"iss":"https://jwt-idp.example.com",
 "prn":"mailto:mike@example.com",
 "aud":"https://jwt-rp.example.net",
 "iat":1300815780,
 "exp":1300819380,
 "http://claims.example.com/member":true}
```

Server Validates the Token

The request is evaluated based of the following information:

- Prn (principal) claim — Username of the user getting the token. If `client_id`, then client is authenticating as itself. If `user`, then it's a regular impersonation flow with `username`, `userid/orgid`, `userurl`.
- Iss (issuer) claim — Consumer key from the remote access.
- Exp (expirations) claim — When the token expires. this limits the time window during which the JWT can be used.
- Aud (audience) claim — contains a URI reference that identifies the authorization server as the intended audience. The authorization server MUST verify that it is an intended audience for the JWT.

Server Sends a Response

After the request is verified, Database.com sends a response to the client. The following parameters are in the body of the response:

- `access_token`—Database.com session ID that can be used with the Web services API.

If the JWT is not valid or has expired, the authorization server constructs an error response. The value of the `error` parameter MUST be the `invalid_grant` error code. The authorization server may include additional information regarding the reasons the JWT was considered invalid using the `error_description` or `error_uri` parameters.

Using the Access Token

User Permissions Needed	
To manage, create, edit, and delete OAuth applications:	“Manage Remote Access”

After a consumer using OAuth version 2.0 has an access token, the method of using the token depends on the API being used.

- For the SOAP API, the access token is placed in the Database.com SOAP authentication header. See [Web Services API Developer's Guide](#).
- For the identity URL, use either an HTTP authorization header (as with the REST API) or use as an HTTP parameter `oauth_token`.

Managing OAuth 2.0 SAML Bearer Flow

OAuth 2.0 SAML Bearer Flow

User Permissions Needed	
To manage, create, edit, and delete OAuth applications:	“Manage Remote Access”

The SAML bearer is a SAML Assertion based token. It is provided within the query string and is base 64 URL encoded. It includes the following sections:

- Subject/NameID that maps to the PRN / Username.
- Issuer element value that is mapped to the ISS as the client id
- Conditions where the NotOnOrAfter attribute is used to validate the expiration date of the assertion.
- Audience that is mapped to the aud and used for validation.

The SAML token has three required parameters:

- `client_id`: The client identifier.
- `client_assertion_type`: The format of the assertion as defined by the authorization server. The value MUST be an absolute URL
- `client_assertion`: The assertion being used to authenticate the client. Specific serialization of the assertion is defined by profile documents. The serialization MUST be encoded for transport within HTTP forms. It is recommended that you use the base64url.

To Implement SAML bearer assertion support:

- The issuer must be the `client_id`
 - ◊ If Subject = `client_id`, then client is authenticating as itself
 - ◊ if Subject = user, then it's a regular impersonation flow (`username`, `userid/orgid`, `userurl`)
- SubjectConfirmation must have the bearer method uri “urn:oasis:names:tc:SAML:2.0:cm:bearer”
- SubjectConfirmationData must have a recipient of the token service being used
- SubjectConfirmationData must allow for only limited clock skew (~5 min)
- AudienceRestriction must be the token service
- The signature must be either validated using the client's provided certificate.
- Follow same rules as existing SAML implementation. No refresh tokens.



Note: Scope is not supported in the flow. If it's a user impersonation, then the scope is that of the previously issued grant. If logging in as a client, then it's the scope of that `client_user`.

The general flow for a SAML bearer token is similar to a refresh token flow within OAuth. It is a way to exchange authentication information after a prior client is authorized by the user. An exchange is done from one SSO provider to another using shared secrets that don't go across the user's browser.

The following are the steps for the SAML token authentication flow.

1. Organization creates a Remote Access OAuth Consumer.
2. Organization writes an application that generates a SAML token.
3. Token is sent in the request to the `/services/oauth2/token`.
4. Token servlet validates signature with the shared secret.
5. Access token is created and sent back to client.

Client Creates a Bearer Token

During this step, the client must create a Remote Access application in an organization so they can get a consumer secret. Using this secret, the client writes an application that generates a SAML token with a private key for an RSA-256 based signature. The public key is uploaded using Remote Access.

Client Sends the Token

The client creates and sends the SAML request to Database.com with the following parameters:

- grant_type: urn:ietf:params:oauth:grant-type:saml2-bearer
- assertion: that is based 64 URL encoded

Here is a sample token request:

```
[I-D.ietf.oauth-v2]
POST /token HTTP/1.1
Host: server.example.com
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code&
code=i1WsRn1uB1&
client_id=s6BhdRkqt3&
client_assertion_type=urn%3Aoasis%3Anames%3Atc%3ASAML%3A2.0%3Aassertion&
client_assertion=PHNhbWxwOl...[omitted for brevity]...ZT
```

Server Validates the Token

The request is evaluated based of the following information:

- prn (principal) claim — Username of the user getting the token. If client_id, then client is authenticating as itself. If user, then it's a regular impersonation flow with username, userid/orgid, userurl.
- Subject/NameID — the username of the user trying to get access.
- Conditions — the expiration date of the token.
- Audience — a URL which is compared to the login for example mydomain or test.salesforce.com.

Server Sends a Response

After the request is verified, Database.com sends a response to the client. Responses for access tokens follow the same format as authorization_code flows and refresh token flows.

If the token is not valid or has expired, the authorization server constructs and error response. The value of the error parameter MUST be the invalid_grant error code. The authorization server may include additional information regarding the reasons the token was considered invalid using the error_description or error_uri parameters. Here is a sample error response:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
Cache-Control: no-store
{
  "error": "invalid_grant",
  "error_description": "Audience validation failed"
}
```

Additional Implementation Details

SAML bearer implementation code is found in the identity-common-api module. The SAML2Bearer uses the SAMLTokenParser to parse the assertion document. The SAMLTokenParser uses Google's Base64 implementation to decode the base64url encoded string. This implementation works across languages types a bit better than the Base64 implementation found in the SFDC code base line. Here are some additional details:

- Signature is validated based on a public key that is uploaded to the Remote Access page.
- Assumes an assertion tag as the root of the document.
- Signature is validated same as all SAML signature validation using XMLSig and comparing ids and duplicate ids to ensure a valid signature. If duplicate IDs are found, it fails signature validation.
- Namespace is not enforced in the bearer implementation to support the document specification.
- Allows users to have a null namespace and still support SAML tags appropriately.

Managing SAML Assertion Flow

SAML Assertion Flow

User Permissions Needed	
To manage, create, edit, and delete OAuth applications:	“Manage Remote Access”

The SAML assertion flow is an alternative for organizations that are currently using SAML to access Database.com using single sign-on, and want to access the Web services API the same way. The SAML assertion flow can only be used inside a single organization. You do not have to create a remote access application to use this assertion flow. Clients can use this to federate with the API using a SAML assertion, in much the same way as they would federate with Database.com for Web single sign-on. See [About Single Sign-On](#) on page 289.

The following are the general steps for using this flow. Many of the steps are described in more detail, below.

1. [Configure SAML](#) on page 345 for your organization. You must use SAML version 2.0.
2. [Exchange a SAML assertion for an access token.](#)
3. Database.com [sends the response](#).

Configuring SAML for OAuth

To configure your organization to use SAML, follow the instructions in [Configuring SAML Settings for Single Sign-On](#). Once you have configured SAML, you can use the exact same configuration for both Web and API federation.

Two URLs are provided after you configure SAML for your organization:

- Database.com Login URL—Use this URL when doing Web single sign-on.
- OAuth 2.0 Token Endpoint—Use this URL when exchanging a SAML assertion for an access token to be used with the API.

When generating SAML assertions to be used with the token endpoint, the recipient URL in the assertion may be the value from either the OAuth 2.0 Token Endpoint or the Database.com Login URL.

Exchange a SAML Assertion for an Access Token

In order to exchange a SAML assertion for an access token, your client must obtain or generate a valid SAML response and POST this to the token endpoint. The method of obtaining this response is up to the client to determine. Once the client has a valid response, it sends the following parameters:

- grant_type—Value must be assertion for this flow.
- assertion—A Base-64 encoded SAML response that would normally be used for Web single sign-on
- assertion_type—Must be urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser
- format—Expected return format. This parameter is optional. The default is json. Values are:
 - ◊ urlencoded
 - ◊ json
 - ◊ xml

The following is the body of an example of an out-of-band POST made to the <https://login.database.com/services/oauth2/token>:

```
grant_type=assertion&assertion_type=
urn%3Aoasis%3Anames%3Atc%3ASAML%3A2.0%3Aprofiles%3ASSO%3Abrowser&
assertion=PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPHNhbw. . .
```

Database.com Server Sends a Response

After the SAML response is verified, Database.com sends a response to the client. The following parameters are in the body of the response:

- `access_token`—Database.com session ID that can be used with the Web services API.
- `id`—Identity URL that can be used to both identify the user as well as query for more information about the user. See [Using Identity URLs](#) on page 282.

The following is an example response from Database.com:

```
{"id":"https://login.database.com/id/00Dx000000BV7z/005x0000012Q9P",
"instance_url":"https://na1.salesforce.com","access_token":
"00Dx000000BV7z!AR8AQNhMmQeDIKR0.hZagSTaEPCKmoXeYnkaxQnqWlG6Sk9U3i3IFjEH
IzDlsYdU0qoVCXNjtPOwdb7u5rKfq9N1dfAKoQjd"}
```

If an error occurs during this step, the response contains an error message with these parts:

- `error`—Error code
- `error_description`—Description of the error with additional information.
 - ◊ `unsupported_response_type`—response type not supported
 - ◊ `invalid_request`—HTTPS required
 - ◊ `invalid_request`—must use HTTP POST
 - ◊ `invalid_assertion_type`—specified assertion type is not supported
 - ◊ `invalid_grant`—invalid authorization code
 - ◊ `invalid_grant`—IP restricted or invalid login hours
 - ◊ `inactive_user`—user is inactive
 - ◊ `inactive_org`—organization is locked, closed, or suspended
 - ◊ `rate_limit_exceeded`—number of logins exceeded

Any login error not listed receives a generic authentication failure with text describing the error. For example, `LOGIN_ERROR_INVALID_PASSWORD` would have the following error response:

```
{"error":"authentication_failure","error_description":"invalid password"}
```

- `error_uri`—A link to the SAML Assertion Validator, which contains more information about the failure. This is only returned when Database.com is able to parse the assertion. See [Validating SAML Settings for Single Sign-On](#) on page 280.

The following is an example error:

```
{"error_uri":"https://na1.salesforce.com/setup/secur/SAMLValidationPage.apexp",
"error":"invalid_grant","error_description":"invalid assertion"}
```

Importing and Exporting Data

Understanding Data Loader

Data Loader Overview

Data Loader is a client application for the bulk import or export of data. Use it to insert, update, delete, or export Database.com records.

When importing data, Data Loader reads, extracts, and loads data from comma separated values (CSV) files or from a database connection. When exporting data, it outputs CSV files.



Note: If commas are not appropriate for your locale, use a tab or other delimiter.

You can use Data Loader in two different ways:

- User interface—When you use the user interface, you work interactively to specify the configuration parameters, CSV files used for import and export, and the field mappings that map the field names in your import file with the field names in Database.com.
- Command line—When you use the command line, you specify the configuration, data sources, mappings, and actions in files. This enables you to set up Data Loader for automated processing.

Data Loader offers the following key features:

- An easy-to-use wizard interface for interactive use
- An alternate command line interface for automated batch operations
- Support for large files with up to 5 million records
- Drag-and-drop field mapping
- Support for all objects, including custom objects
- Detailed success and error log files in CSV format
- A built-in CSV file viewer
- Support for Windows 7

To get started, see the following topics:

- [When to Use Data Loader](#)
- [Installing Data Loader](#)



Note: In previous versions, Data Loader has been known as “AppExchange Data Loader” and “Sforce Data Loader.”

When to Use Data Loader

Refer to the following guidelines to determine when to use Data Loader:

Use Data Loader when:

- You need to load 50,000 to 5,000,000 records. If you need to load more than 5,000,000 records, we recommend you work with a Salesforce.com partner.
- You want to schedule regular data loads, such as nightly imports.
- You want to export your data for backup purposes.

Data Types Supported by Data Loader

Data Loader supports the following data types:

Boolean

- True values (case insensitive) = yes, y, true, on, 1
- False values (case insensitive) = no, n, false, off, 0

Date Formats

We recommend you specify dates in the format *yyyy-MM-ddTHH:mm:ss.SSS+/-HHmm:*

- *yyyy* is the four-digit year
- *MM* is the two-digit month (01-12)
- *dd* is the two-digit day (01-31)
- *HH* is the two-digit hour (00-23)
- *mm* is the two-digit minute (00-59)
- *ss* is the two-digit seconds (00-59)
- *sss* is the three-digit milliseconds (000-999)
- *+/-HHmm* is the Zulu (UTC) time zone offset

The following date formats are also supported:

- *yyyy-MM-dd'T'HH:mm:ss.SSS'Z'*
- *yyyy-MM-dd'T'HH:mm:ss.SSS Pacific Standard Time*
- *yyyy-MM-dd'T'HH:mm:ss.SSSPacific Standard Time*
- *yyyy-MM-dd'T'HH:mm:ss.SSS PST*
- *yyyy-MM-dd'T'HH:mm:ss.SSSPST*
- *yyyy-MM-dd'T'HH:mm:ss.SSS GMT-08:00*
- *yyyy-MM-dd'T'HH:mm:ss.SSSGMT-08:00*
- *yyyy-MM-dd'T'HH:mm:ss.SSS -800*
- *yyyy-MM-dd'T'HH:mm:ss.SSS-800*
- *yyyy-MM-dd'T'HH:mm:ss*
- *yyyy-MM-dd HH:mm:ss*
- *yyyyMMdd'T'HH:mm:ss*
- *yyyy-MM-dd*
- *MM/dd/yyyy HH:mm:ss*
- *MM/dd/yyyy*

Note the following tips for date formats:

- To enable date formats that begin with the day rather than the month, select the Use European date format box in the Settings dialog. European date formats are dd/MM/yyyy and dd/MM/yyyy HH:mm:ss.
- If your computer's locale is east of Greenwich Mean Time (GMT), we recommend that you change your computer setting to GMT in order to avoid date adjustments when inserting or updating records. A knowledge base solution for this issue is available in the Help & Training window; search for "data loader date."

- Only dates within a certain range are valid. The earliest valid date is 1700-01-01T00:00:00Z GMT, or just after midnight on January 1, 1700. The latest valid date is 4000-12-31T00:00:00Z GMT, or just after midnight on December 31, 4000.



Note: These values are offset by your time zone. For example, in the Pacific time zone, the earliest valid date is 1699-12-31T16:00:00, or 4:00 PM on December 31, 1699.

Double

Standard double string

ID

A Database.com ID is a case-sensitive 15-character or case-insensitive 18-character alphanumeric string that uniquely identifies a particular record.



Tip: To ensure data quality, make sure that all Database.com IDs you enter in Data Loader are in the correct case.

Integer

Standard integer string

String

All valid XML strings; invalid XML characters are removed.

Data Loader Third-Party Licenses

The following third-party licenses are included with the installation of Data Loader:

Technology	Version Number	License
Apache Jakarta Commons BeanUtils	1.6	http://www.apache.org/licenses/LICENSE-2.0
Apache Commons Collections	3.1	http://www.apache.org/licenses/LICENSE-2.0
Apache Commons Database Connection Pooling (DBCP)	1.2.1	http://www.apache.org/licenses/LICENSE-2.0
Apache Commons Logging	1.0.3	http://www.apache.org/licenses/LICENSE-1.1
Apache Commons Object Pooling Library	1.2	http://www.apache.org/licenses/LICENSE-2.0
Apache Log4j	1.2.8	http://www.apache.org/licenses/LICENSE-2.0
Eclipse SWT	3.452	http://www.eclipse.org/legal/epl-v10.html
OpenSymphony Quartz Enterprise Job Scheduler	1.5.1	http://www.opensymphony.com/quartz/license.action
Rhino JavaScript for Java	1.6R2	http://www.mozilla.org/MPL/MPL-1.1.txt

Technology	Version Number	License
Spring Framework	1.2.6	http://www.apache.org/licenses/LICENSE-2.0.txt

 **Note:** Salesforce.com is not responsible for the availability or content of third-party websites.

Spring Framework Overview

The Data Loader configuration files are based on the [Spring Framework](#), which is an open source full-stack Java/J2EE application framework.

The Spring Framework allows you to use XML files to configure beans. Each bean represents an instance of an object; the parameters correspond to each object's setter methods. A typical bean has the following attributes:

id

Uniquely identifies the bean to `XmlBeanFactory`, which is the class that gets objects from an XML configuration file.

class

Specifies the implementation class for the bean instance.

For more information on the Spring Framework, see [the official documentation](#) and the [support forums](#). Note that salesforce.com cannot guarantee the availability or accuracy of external websites.

Installing, Configuring, and Uninstalling Data Loader

Installing Data Loader

User Permissions Needed	
To access the page to download Data Loader:	“Modify All Data”
To use Data Loader:	The appropriate user permission for the operation you are doing, for example, “Create” on Merchandise__c to insert new merchandise custom objects

System Requirements

To use Data Loader, you need:

- Windows 7 or Windows XP
- 90 MB free disk space
- 256 MB available memory
- Java JRE 1.5 or later (Windows 7 or Windows XP)
- Sun JVM 1.5 or later (Windows 7 or Windows XP)

Installation Procedure



Caution: Over time, multiple versions of Data Loader client application have been available for download. Different versions have different entries in the Add or Remove Programs dialog in your Windows Control Panel. Some versions were released with earlier product names such as “AppExchange Data Loader” or “Sforce Data Loader.” You can run *different* versions at the same time on one computer. However, do not install multiple copies of the same version.

The latest version is always available from Database.com at **Data Management > Data Loader**. If you have previously installed the latest version and want to install it again, first remove it from your computer by using the Add or Remove Programs dialog in Windows Control Panel.

1. In the application, click **Data Management > Data Loader**.
2. Click **Download the Data Loader** and save the installer to your PC. If you're prompted to run or save the file, click **Run**. If you're then prompted to allow the program to make changes to the computer, click **Yes**.
3. Double-click the downloaded file to launch the InstallShield wizard.
4. Click **Next**.
5. Accept the license agreement and click **Next**.
6. Accept the default installation directory, or click **Change...** to choose another directory. Click **Next**.
7. Click **Install**.
8. Click **Finish**.
9. To start the Data Loader, double-click the Data Loader icon on your desktop, or choose **Start > All Programs > salesforce.com > Apex Data Loader > Apex Data Loader**.



Tip:

If you experience login issues in the command line interface after upgrading to a new version of Data Loader, please try re-encrypting your password to solve the problem. For information on the password encryption utility, see [Encrypting From the Command Line](#) on page 365.

If you want to download the source code and make changes, an open source version of Data Loader is available at <https://github.com/forcedotcom/dataloader>.

Login Considerations

If your organization [restricts IP addresses](#), logins from untrusted IPs are blocked until they're activated. Database.com automatically sends you an activation email that you can use to log in. The email contains a security token that you must add to the end of your password. For example, if your password is `mypassword`, and your [security token](#) is `XXXXXXXXXXXX`, you must enter `mypasswordXXXXXXXXXXXX` to log in.

Configuring Data Loader

Use the Settings menu to change the default operation settings of Data Loader.

1. Start Data Loader by choosing **Start > Programs > salesforce.com > Data Loader > Data Loader**.
2. Choose **Settings > Settings**.
3. Edit the fields as desired:

Field	Description
Batch size	In a single insert, update, upsert, or delete operation, records moving to or from Database.com are processed in increments of this size. The maximum value is 200. We recommend a value between 50 and 100.

Field	Description
	The maximum value is 10,000 if the <code>Use Bulk API</code> option is selected.
<code>Insert null values</code>	Select this option to insert blank mapped values as <code>null</code> values during data operations. Note that when you are updating records, this option instructs Data Loader to overwrite any existing data in mapped fields.
	This option is not available if the <code>Use Bulk API</code> option is selected. Empty field values are ignored when you update records using the Bulk API. To set a field value to <code>null</code> when the <code>Use Bulk API</code> option is selected, use a field value of <code>#N/A</code> .
<code>Server host</code>	Enter the URL of the Database.com server with which you want to communicate. For example, if you are loading data into a test database , change the URL to <code>https://test.database.com</code> .
<code>Reset URL on Login</code>	By default, Database.com resets the URL after login to the one specified in <code>Server host</code> . To turn off this automatic reset, disable this option.
<code>Compression</code>	Compression enhances the performance of Data Loader and is turned on by default. You may want to disable compression if you need to debug the underlying SOAP messages. To turn off compression, enable this option.
<code>Timeout</code>	Specify how many seconds Data Loader waits to receive a response back from the server before returning an error for the request.
<code>Query request size</code>	In a single export or query operation, records are returned from Database.com in increments of this size. The maximum value is 2,000 records. Larger values may improve performance but use more memory on the client.
<code>Generate status files for exports</code>	Select this option to generate success and error files when exporting data.
<code>Read all CSVs with UTF-8 encoding</code>	Select this option to force files to open in UTF-8 encoding, even if they were saved in a different format.
<code>Write all CSVs with UTF-8 encoding</code>	Select this option to force files to be written in UTF-8 encoding.
<code>Use European date format</code>	Select this option to support the date formats <code>dd/MM/yyyy</code> and <code>dd/MM/yyyy HH:mm:ss</code> .
<code>Allow field truncation</code>	Select this option to truncate data in the following types of fields when loading that data into Database.com: Email,

Field	Description
	<p>Multi-select Picklist, Phone, Picklist, Text, and Text (Encrypted).</p>
	<p>In Data Loader versions 14.0 and earlier, values for fields of those types are truncated by Data Loader if they are too large. In Data Loader version 15.0 and later, the load operation fails if a value is specified that is too large.</p>
	<p>Selecting this option allows you to specify that the previous behavior, truncation, be used instead of the new behavior in Data Loader versions 15.0 and later. This option is selected by default and has no effect in versions 14.0 and earlier.</p>
	<p>This option is not available if the <code>Use Bulk API</code> option is selected. In that case, the load operation fails for the row if a value is specified that is too large for the field.</p>
<code>Use Bulk API</code>	<p>Select this option to use the Bulk API to insert, update, upsert, delete, and hard delete records. The Bulk API is optimized to load or delete a large number of records asynchronously. It's faster than the default SOAP-based API due to parallel processing and fewer network round-trips.</p>
	<p> Caution: When you select the Hard Delete operation, the deleted records are not stored in the Recycle Bin. Instead, they become immediately eligible for deletion. The permission for this operation, Bulk API Hard Delete, is disabled by default and must be enabled by an administrator. A user license is required for hard delete.</p>
	<p>For more information, see Configuring the Data Loader to Use the Bulk API on page 359 and “Data Loader Behavior with Bulk API Enabled” in the online help</p>
<code>Enable serial mode for Bulk API</code>	<p>Select this option to use serial instead of parallel processing for Bulk API. Processing in parallel can cause database contention. When this is severe, the load may fail. Using serial mode guarantees that batches are processed one at a time. Note that using this option may significantly increase the processing time for a load.</p>
	<p>This option is only available if the <code>Use Bulk API</code> option is selected.</p>

Field	Description
Time Zone	Select this option to specify a default time zone.
	If a date value does not include a time zone, this value is used.
	<ul style="list-style-type: none"> • If no value is specified, the time zone of the computer where Data Loader is installed is used. • If an incorrect value is entered, GMT is used as the time zone and this fact is noted in the Data Loader log.
	<p>Valid values are any time zone identifier which can be passed to the Java <code>getTimeZone (java.lang.String)</code> method. The value can be a full name such as America/Los_Angeles, or a custom ID such as GMT-8:00.</p>
Proxy host	The host name of the proxy server, if applicable.
Proxy port	The proxy server port.
Proxy username	The username for proxy server authentication.
Proxy password	The password for proxy server authentication.
Proxy NTLM domain	The name of the Windows domain used for NTLM authentication.
Start at row	If your last operation failed, you can use this setting to begin where the last successful operation finished.

4. Click **OK** to save your settings.

Uninstalling the Data Loader

To uninstall the Data Loader client application:

1. Go to **Start > Control Panel > Add or Remove Programs**.
2. Select the Data Loader program.
3. Click **Remove**. The uninstaller removes the program from your computer.

Working with Data Loader from the User Interface

Inserting, Updating, and Deleting Data

Inserting, Updating, or Deleting Data Using Data Loader

User Permissions Needed	
To insert records:	“Create” on the record
To update records:	“Edit” on the record
To upsert records:	“Create” or “Edit” on the record
To delete records:	“Delete” on the record
To hard delete records	“Delete” on the record

The insert, update, upsert, delete, and hard delete wizards in Data Loader allow you to add new records, modify existing records, or delete existing records. Note that “upsert” is a combination of inserting and updating - if a record in your file matches an existing record, the existing record is updated with the values in your file. If no match is found, then the record is created as new. When you hard delete records, the deleted records are not stored in the Recycle Bin and become immediately eligible for deletion. For more information, see [Configuring Data Loader](#) on page 351.

1. Start Data Loader by choosing **Start > Programs > salesforce.com > Data Loader > Data Loader**.
2. Click **Insert, Update, Upsert, Delete** or **Hard Delete**. These commands can also be found in the File menu.
3. Enter your Database.com username and password. Click **Log in** to log in. After your login completes successfully, click **Next**. (Until you log out or close the program, you are not asked to log in again.)

If your organization [restricts IP addresses](#), logins from untrusted IPs are blocked until they're activated. Database.com automatically sends you an activation email that you can use to log in. The email contains a security token that you must add to the end of your password. For example, if your password is mypassword, and your [security token](#) is XXXXXXXXXXXX, you must enter mypasswordXXXXXXXXXX to log in.

4. Choose an object. For example, if you are inserting Merchandise__c custom object records, select **Merchandise__c**. If your object name does not display in the default list, check **Show all objects** to see a complete list of the objects that you can access. The objects are listed by localized label name, with developer name noted in parentheses. For object descriptions, see the [Database.com Object Reference](#).
5. Click **Browse...** to select your CSV file. For example, if you are inserting Merchandise__c records, you could specify a CSV file named `insertmerchandise.csv` containing a Name column for the names of the new merchandise records.
6. Click **Next**. After the object and CSV file are initialized, click **OK**.
7. If you are performing an upsert:
 - a. Your CSV file must contain a column of ID values for matching against existing records. The column may be either an external ID (a custom field with the “External ID” attribute), or `Id` (the Database.com record ID). From the drop-down list, select which field to use for matching. If the object has no external ID fields, `Id` is automatically used. For more information on external IDs, see [Custom Field Attributes](#) on page 101. Click **Next** to continue.
 - b. If your file includes the external IDs of an object that has a relationship to your chosen object, enable that external ID for record matching by selecting its name from the drop-down list. If you make no selection here, you can use the related object's `Id` field for matching by mapping it in the next step. Click **Next** to continue.

8. Define how the columns in your CSV file map to Database.com fields. Click **Choose an Existing Map** to select an existing field mapping, or click **Create or Edit a Map** to create a new map or modify an existing map. For more details and an example of usage, see [Defining Field Mappings](#) on page 356.
9. Click **Next**.
10. For every operation, the Data Loader generates two unique CSV log files; one file name starts with “success,” while the other starts with “error.” Click **Browse...** to specify a directory for these files.
11. Click **Finish** to perform the operation, and then click **Yes** to confirm.
12. As the operation proceeds, a progress information window reports the status of the data movement.
13. After the operation completes, a confirmation window summarizes your results. Click **View Successes** to view your success file, click **View Errors** to open your errors file, or click **OK** to close. For more information, see [Reviewing Output Files](#) on page 359.



Tip:

- If you are updating or deleting large amounts of data, review [Performing Mass Updates](#) and [Performing Mass Deletes](#) for tips and best practices.
- There is a five-minute limit to process 100 records when the Bulk API is enabled. Also, if it takes longer than 10 minutes to process a file, the Bulk API places the remainder of the file back in the queue for later processing. If the Bulk API continues to exceed the 10-minute limit on subsequent attempts, the file is placed back in the queue and reprocessed up to 10 times before the operation is permanently marked as failed. Even if the processing failed, some records could have completed successfully, so you must check the results. If you get a timeout error when loading a file, split your file into smaller files, and try again.

Defining Field Mappings

Whenever you insert, delete, or update files, use the Mapping Dialog window to associate Database.com fields with the columns of your CSV file. For more information, see [Inserting, Updating, or Deleting Data Using Data Loader](#) on page 355.

1. To automatically match fields with columns, click **Auto-Match Fields to Columns**. The Data Loader automatically populates the list at the bottom of the window, based on the similarity of field and column names. Note that for a delete operation, automatic matching works only on the ID field.
2. To manually match fields with columns, click and drag fields from the list of Database.com fields at the top to the list of CSV column header names at the bottom. For example, if you are inserting new custom object records where the object has a custom field named `Description__c` and your CSV file contains data for this field in a column named `Descriptions`, click and drag the `Description__c` field to the right of the `Descriptions` column header field.
3. Optionally, click **Save Mapping** to save this mapping for future use. Specify a name for the SDL mapping file.
If you select an existing file, the contents of that file are replaced. Click **Yes** to confirm this action, or click **No** to choose another file.
4. Click **OK** to use your mapping for the current operation.

Performing Mass Deletes

To delete a large number of records at one time using the Data Loader, we recommend the following steps:

1. As a backup measure, export the objects you wish to delete, being sure to select all fields. (See [Exporting Data](#) on page 358.) Save an extra copy of the generated CSV file.
2. Next, export the objects you wish to delete, this time using only the record ID as the desired criteria.
3. Launch the Data Loader and follow the delete or hard delete wizard. Map only the ID column. See [Inserting, Updating, or Deleting Data Using Data Loader](#) on page 355.

- After the operation, review your success and error log files. See [Reviewing Output Files](#) on page 359.

Performing Mass Updates

To update a large number of records at one time, we recommend the following steps:

- Obtain your data by performing an export of the objects you wish to update. See [Exporting Data](#) on page 358.
- As a backup measure, save an extra copy of the generated CSV file.
- Open your working file in a CSV editor such as Excel, and update your data.
- Launch the Data Loader and follow the update wizard. Note that matching is done according to record ID. See [Inserting, Updating, or Deleting Data Using Data Loader](#) on page 355.
- After the operation, review your success and error log files. See [Reviewing Output Files](#) on page 359.
- If you made a mistake, use the backup file to update the records to their previous values.

Reviewing Output Files

After every import or export, Data Loader generates two CSV output files that contain the results of the operation. One file name starts with “success,” while the other starts with “error.” During every export, Data Loader saves the extracted data to a CSV file that you specify in the wizard. Data Loader has a built-in CSV file viewer with which you can open and view these files.

To view output files from a Data Loader operation:

- Choose **View > View CSV**.
- Specify the number of rows to view. Each row in the CSV file corresponds to one Database.com record. The default is 1000.
- To view a CSV file of your choice, click **Open CSV**. To view the last success file, click **Open Success**. To view the last error file, click **Open Error**. The CSV file opens in a new window.
- Optionally, click **Open in External Program** to open the file in the associated external program, such as Microsoft® Office Excel.

The “success” file contains all of the records that were successfully loaded. In this file, there's a column for the newly generated record IDs. The “error” file contains all of the records that were rejected from the load operation. In this file, there's a column that describes why the load failed.

- Click **Close** to return to the CSV Chooser window, and then click **OK** to exit the window.

 **Note:** To generate success files when exporting data, select the `Generate status files for exports` setting. For more information, see [Configuring Data Loader](#) on page 351.

Troubleshooting Data Loader Operations

If you need to investigate a problem with the Data Loader, or if requested by salesforce.com Customer Support, you can access log files that track the operations and network connections made by the Data Loader. The two log files are:

sdl.log

Contains a detailed chronological list of Data Loader log entries. Log entries marked “INFO” are procedural items, such as logging in to Database.com. Log entries marked “ERROR” are problems such as a submitted record missing a required field.

sdl_out.log

A supplemental log that contains additional information not captured in sdl.log. For example, it includes log entries for the creation of proxy server network connections.

These files can be opened with commonly available text editor programs, such as Microsoft Notepad.

You can quickly open these files by entering %TEMP%\sdl.log and %TEMP%\sdl_out.log in either the Run dialog or the Windows Explorer address bar.

If you are having login issues from the command line, ensure that the password provided in the configuration parameters is encrypted. If you are having login issues from the UI, you may need to obtain a new security token.

Exporting Data

Exporting Data

User Permissions Needed	
To export records:	“Read” on the records
To export all records:	“Read” on the records

You can use the Data Loader export wizard to extract data from any Database.com object. When you export, you can choose to include (**Export All**) or exclude (**Export**) soft-deleted records.

1. Start the Data Loader by choosing **Start > Programs > salesforce.com > Data Loader > Data Loader**.
2. Click **Export** or **Export All**. These commands can also be found in the File menu.
3. Enter your Database.com username and password. Click **Log in** to log in. After your login completes successfully, click **Next**. (Until you log out or close the program, you will not be asked to log in again.)

If your organization [restricts IP addresses](#), logins from untrusted IPs are blocked until they're activated. Database.com automatically sends you an activation email that you can use to log in. The email contains a security token that you must add to the end of your password. For example, if your password is mypassword, and your [security token](#) is XXXXXXXXXXXX, you must enter mypasswordXXXXXXXXXX to log in.

4. Choose an object. If your object name does not display in the default list, check `Show all objects` to see a complete list of objects that you can access. The objects will be listed by localized label name, with developer name noted in parentheses. For object descriptions, see the [Web Services API Developer's Guide](#).

5. Click **Browse...** to select the CSV file to which the data will be exported. You can enter a new file name to create a new file or choose an existing file.

If you select an existing file, the contents of that file are replaced. Click **Yes** to confirm this action, or click **No** to choose another file.

6. Click **Next**.
7. Create a SOQL query for the data export. SOQL is the Salesforce Object Query Language that allows you to construct simple but powerful query strings. Similar to the SELECT command in SQL, SOQL allows you to specify the source object, a list of fields to retrieve, and conditions for selecting rows in the source object.
 - a. Choose the fields you want to export.
 - b. Optionally, select conditions to filter your data set. If you do not select any conditions, all the data to which you have read access will be returned.
 - c. Review the generated query and edit if necessary.



Tip: You can use a SOQL relationship query to include fields from a related object. For example:

```
Select Units_Sold__c, Unit_Price__c, Merchandise__r.Description__c FROM Line_Item__c
```

When using relationship queries in the Data Loader, the fully specified field names are case-sensitive. For example, using MERCHANTISE__R.DESCRIPTION__C instead of Merchandise__r.Description__c does not work.

The Data Loader does not support nested queries or querying child objects. For example, queries similar to the following return an error:

```
SELECT Name, (SELECT Name FROM Line_Items__r) FROM Merchandise__c
```

For more information on SOQL, see the [Web Services API Developer's Guide](#).

8. Click **Finish**, then click **Yes** to confirm.
9. A progress information window reports the status of the operation.
10. After the operation completes, a confirmation window summarizes your results. Click **View Extraction** to view the CSV file, or click **OK** to close. For more details, see [Reviewing Output Files](#) on page 359.

Reviewing Output Files

After every import or export, Data Loader generates two CSV output files that contain the results of the operation. One file name starts with “success,” while the other starts with “error.” During every export, Data Loader saves the extracted data to a CSV file that you specify in the wizard. Data Loader has a built-in CSV file viewer with which you can open and view these files.

To view output files from a Data Loader operation:

1. Choose **View > View CSV**.
2. Specify the number of rows to view. Each row in the CSV file corresponds to one Database.com record. The default is 1000.
3. To view a CSV file of your choice, click **Open CSV**. To view the last success file, click **Open Success**. To view the last error file, click **Open Error**. The CSV file opens in a new window.
4. Optionally, click **Open in External Program** to open the file in the associated external program, such as Microsoft® Office Excel.

The “success” file contains all of the records that were successfully loaded. In this file, there’s a column for the newly generated record IDs. The “error” file contains all of the records that were rejected from the load operation. In this file, there’s a column that describes why the load failed.

5. Click **Close** to return to the CSV Chooser window, and then click **OK** to exit the window.



Note: To generate success files when exporting data, select the **Generate status files for exports** setting. For more information, see [Configuring Data Loader](#) on page 351.

Configuring the Data Loader to Use the Bulk API

The Bulk API is optimized to load or delete a large number of records asynchronously. It is faster than the SOAP-based API due to parallel processing and fewer network round-trips. By default, Data Loader uses the SOAP-based API to process records.

To configure Data Loader to use the Bulk API for inserting, updating, upserting, deleting, and hard deleting records:

1. Start Data Loader by choosing **Start > Programs > salesforce.com > Data Loader > Data Loader**.
2. Choose **Settings > Settings**.
3. Select the `Use Bulk API` option.
4. Click **OK**.



Note: You can also select the `Enable serial mode for Bulk API` option. Processing in parallel can cause database contention. When this is severe, the load may fail. Using serial mode guarantees that batches are processed one at a time. Note that using this option may significantly increase the processing time for a load.



Caution: When you select the **Hard Delete** operation, the deleted records are not stored in the Recycle Bin. Instead, they become immediately eligible for deletion. The permission for this operation, Bulk API Hard Delete, is disabled by default and must be enabled by an administrator. A user license is required for hard delete.

Working with Data Loader from the Command Line

Command Line Quick Start Introduction



[Watch a Demo \(5:21 minutes\)](#)

In addition to using Data Loader interactively to import and export data, you can also run it from the command line. This enables you to automate the import and export of data.

This quick start shows you how to use the Data Loader command line functionality to import data. You'll follow these steps:

- [Step 1: Create the encryption key](#)
- [Step 2: Create the encrypted password for your login username](#)
- [Step 3: Create the Field Mapping File](#)
- [Step 4: Create a `process-conf.xml` file that contains the import configuration settings](#)
- [Step 5: Run the process and import the data](#)

Prerequisites

To step through this quick start, you should have the following:

- Data Loader installed on the computer that runs the command line process.
- The Java Runtime Environment (JRE) installed on the computer that runs the command line process.
- Familiarity with importing and exporting data by using the Data Loader interactively through the user interface. This makes it easier to understand how the command line functionality works.



Tip: When you install Data Loader, sample files are installed in the samples directory. This directory is found below the program directory, for example, `C:\Program Files (x86)\salesforce.com\Apex Data Loader 22.0\samples\`. Examples of files that are used in this quick start can be found in the `\samples\conf` directory.

Step One: Create the Encryption Key

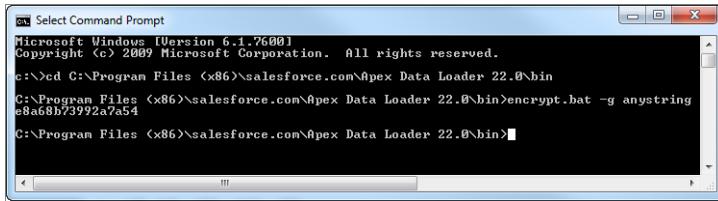
When you use Data Loader from the command line, there's no user interface. Therefore, you'll need to provide the information that you would normally enter in the user interface by using a text file named `process-conf.xml`. For example, you'll need to add to this file the username and password that Data Loader uses to log in to Database.com. The password must be encrypted before you add it to the `process-conf.xml` file, and creating the key is the first step in that process.

1. Open a command prompt window by clicking **Start > All Programs > Accessories > Command Prompt**. Alternatively, you can click **Start > Run**, enter cmd in the **Open** field, and click **OK**.
2. In the command window enter cd\ to navigate to the root directory of the drive where Data Loader is installed.
3. Navigate to the Data Loader \bin directory by entering this command. Be sure to replace the file path with the path from your system.

```
cd C:\Program Files (x86)\salesforce.com\Apex Data Loader 22.0\bin
```

4. Create an encryption key by entering the following command. Replace <seedtext> with any string.

```
encrypt.bat -g <seedtext>
```



Note: To see a list of command-line options for encrypt.bat, type encrypt.bat from the command line.

5. Copy the generated key from the command window to a text file named key.txt and make a note of the file path. In this example, the generated key is e8a68b73992a7a54.



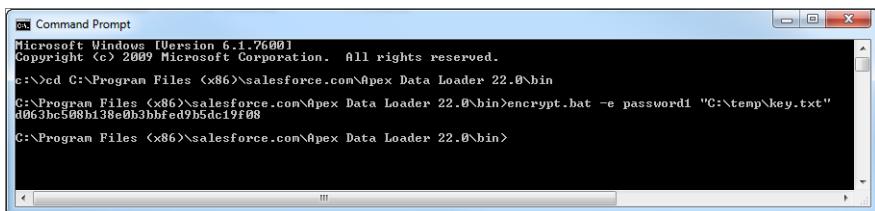
Note: Enabling quick edit mode on a command window can make it easier to copy data to and from the window. To enable quick edit mode, right-click the top of the window and select **Properties**. On the **Options** tab, select **QuickEdit Mode**.

Step Two: Create the Encrypted Password

In this step, you'll create the encrypted password using the key that you generated in the previous step.

1. In the same command prompt window, enter the following command. Replace <password> with the password that Data Loader uses to log in to Database.com. Replace <filepath> with the file path to the key.txt file that you created in the previous step.

```
encrypt.bat -e <password> "<filepath>\key.txt"
```



2. Copy the encrypted password that is generated by the command. You'll use this value in a later step.

Step Three: Create the Field Mapping File

The field mapping file associates data sources with destinations. This is a text file, typically with an .sdl file extension.

1. Copy the following to a text file and save it with a name of accountInsertMap.sdl. This is a data insert so the data source is on the left of the equals sign and the destination field is on the right.

```
#Mapping values
#Thu May 26 16:19:33 GMT 2011
Name=Name
NumberOfEmployees=NumberOfEmployees
Industry=Industry
```



Tip: For complex mappings, you can use the Data Loader user interface to map source and destination fields and then save those mappings to an .sdl file. This is done on the Mapping dialog box by clicking **Save Mapping**.

Step Four: Create the Configuration File

The process-conf.xml file contains the information that Data Loader needs to process the data. Each <bean> in the process-conf.xml file refers to a single process such as an insert, upsert, export, and so on. Therefore, this file can contain multiple processes. In this step, you'll edit the file to insert accounts into Database.com.

1. Make a copy of the process-conf.xml file from the \samples\conf directory. Be sure to maintain a copy of the original because it contains examples of other types of Data Loader processing such as upserts and exports.
2. Open the file in a text editor and replace the contents with the following XML:

```
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"
"http://www.springframework.org/dtd/spring-beans.dtd">
<beans>
    <bean id="accountInsert"
        class="com.salesforce.dataloader.process.ProcessRunner"
        singleton="false">
        <description>accountInsert job gets the account record from the CSV file
            and inserts it into Salesforce.</description>
        <property name="name" value="accountInsert"/>
        <property name="configOverrideMap">
            <map>
                <entry key="sfdc.debugMessages" value="true"/>
                <entry key="sfdc.debugMessagesFile"
                    value="C:\DLTest\Log\accountInsertSoapTrace.log"/>
                <entry key="sfdc.endpoint" value="https://servername.salesforce.com"/>
                <entry key="sfdc.username" value="admin@Org.org"/>
                <!--Password below has been encrypted using key file,
                    therefore, it will not work without the key setting:
                    process.encryptionKeyFile.
                    The password is not a valid encrypted value,
                    please generate the real value using the encrypt.bat utility -->
                <entry key="sfdc.password" value="e8a68b73992a7a54"/>
                <entry key="process.encryptionKeyFile"
                    value="C:\DLTest\Command Line\Config\key.txt"/>
                <entry key="sfdc.timeoutSecs" value="600"/>
                <entry key="sfdc.loadBatchSize" value="200"/>
                <entry key="sfdc.entity" value="Account"/>
                <entry key="process.operation" value="insert"/>
                <entry key="process.mappingFile"
                    value="C:\DLTest\Command Line\Config\accountInsertMap.sdl"/>
                <entry key="dataAccess.name"
                    value="C:\DLTest\In\insertAccounts.csv"/>
                <entry key="process.outputSuccess"
                    value="c:\DLTest\Log\accountInsert_success.csv"/>
                <entry key="process.outputError"
                    value="c:\DLTest\Log\accountInsert_error.csv"/>
            </map>
        </property>
    </bean>
</beans>
```

```

        <entry key="dataAccess.type" value="csvRead"/>
        <entry key="process.initialLastRunDate"
               value="2005-12-01T00:00:00.000-0800"/>
    </map>
</property>
</bean>

```

3. Modify the following parameters in the `process-conf.xml` file. For more information about the process configuration parameters, see [Data Loader Process Configuration Parameters](#) on page 370.

- `sfdc.endpoint`—Enter the URL of the Database.com instance for your organization; for example, <https://na1.salesforce.com>.
- `sfdc.username`—Enter the username Data Loader uses to log in.
- `sfdc.password`—Enter the encrypted password value that you created in step 2.
- `process.mappingFile`—Enter the path and file name of the mapping file.
- `dataAccess.Name`—Enter the path and file name of the data file that contains the accounts that you want to import.
- `sfdc.debugMessages`—Currently set to `true` for troubleshooting. Set this to `false` after your import is up and running.
- `sfdc.debugMessagesFile`—Enter the path and file name of the command line log file.
- `process.outputSuccess`—Enter the path and file name of the success log file.
- `process.outputError`—Enter the path and file name of the error log file.



Caution: Use caution when using different XML editors to edit the `process-conf.xml` file. Some editors add XML tags to the beginning and end of the file which will cause the import to fail.

Step Five: Import the Data

User Permissions Needed	
To insert records:	“Create” on the record
To update records:	“Edit” on the record
To upsert records:	“Create” or “Edit” on the record
To delete records:	“Delete” on the record
To hard delete records:	“Delete” on the record

Now that all the pieces are in place, you can run Data Loader from the command line and insert some new accounts.

1. Copy the following data to a file name `accountInsert.csv`. This is the account data that you'll import into your organization.

```

Name,Industry,NumberOfEmployees
Dickenson plc,Consulting,120
GenePoint,Biotechnology,265
Express Logistics and Transport,Transportation,12300
Grand Hotels & Resorts Ltd,Hospitality,5600

```

2. In the command prompt window, enter the following command:

```
process.bat "<file path to process-conf.xml>" <process name>
```

- Replace <file path to process-conf.xml> with the path to the directory containing process-conf.xml.
- Replace <process name> with the process specified in process-conf.xml.

Your command should look something like this:

```
process.bat "C:\DLTest\Command Line\Config" accountInsert
```

After the process runs, the command prompt window displays success and error messages. You can also check the log files: insertAccounts_success.csv and insertAccounts_error.csv. After the process runs successfully, the insertAccounts_success.csv file will contain the records that you imported along with the ID and status of each record. For more information about the status files, see [Reviewing Output Files](#) on page 359

Understanding Installed Directories and Files

In versions 8.0 and later, [installing the Data Loader](#) creates several directories under the installation directory. The following directories are involved in running the program from the command line for automated batch processing:

bin

Contains the batch files encrypt.bat for [encrypting passwords](#) and process.bat for [running batch processes](#).

For information on running the Data Loader from the command line, see [Using the Command Line Interface](#) on page 366.

conf

The default configuration directory. Contains the configuration files config.properties, Loader.class, and log-conf.xml.

The config.properties file that is generated when you modify the Settings dialog in the graphical user interface is located at C:\Documents and Settings\your Windows username\Application Data\salesforce.com\Data Loader version_number. You can copy this file to the conf installation directory to use it for batch processes.

samples

Contains subdirectories of sample files for reference.

File Path Convention

The file paths provided in these topics start one level below the installation directory. For example, \bin means C:\Program Files \salesforce.com\Data Loader version_number\bin, provided you accepted the default installation directory. If you installed the program to a different location, please substitute that directory path as appropriate.

Data Access Objects

When running the Data Loader in batch mode from the command line, several data access objects are supported. A data access object allows access to an external data source outside of Database.com. They can implement a read interface (DataReader), a write interface (DataWriter), or both. See the following list of object names and descriptions.

csvRead

Allows the reading of a comma or tab-delimited file. There should be a header row at the top of the file that describes each column.

csvWrite

Allows writing to a comma or tab-delimited file. A header row is added to the top of the file based on the column list provided by the caller.

databaseRead

Allows the reading of a database. Use `database-conf.xml` to configure database access. For more information, see [Configuring Database Access](#) on page 368.

databaseWrite

Allows writing to a database. Use `database-conf.xml` to configure database access. For more information, see [Configuring Database Access](#) on page 368.

Encrypting From the Command Line

When running the Data Loader in batch mode from the command line, you must encrypt the following configuration parameters:

- `sfdc.password`
- `sfdc.proxyPassword`

Use the utility described below to perform encryption.

Using the Encryption Utility

The Data Loader offers an encryption utility to secure passwords specified in configuration files. To use the encryption utility:

1. Run `\bin\encrypt.bat`.
2. At the command line, follow the prompts provided to execute the following actions:

Generate a key

Key text is generated onscreen from the text you provide. Carefully copy the key text to a key file, omitting any leading or trailing spaces. The key file can then be used for encryption and decryption.

Encrypt text

Generates an encrypted version of a password or other text. Optionally, you can provide a key file for the encryption. In the configuration file, make sure that the encrypted text is copied precisely and the key file is mentioned.

Verify encrypted text

Given encrypted and decrypted versions of a password, verifies whether the encrypted password provided matches its decrypted version. A success or failure message is printed to the command line.

Upgrading Your Batch Mode Interface

The batch mode interface in versions 8.0 and later of the Data Loader are not backwards-compatible with earlier versions. If you are using a version earlier than 8.0 to run batch processes, your options are as follows:

Maintain the old version for batch use

Do not uninstall your old version of the Data Loader. Continue to use that version for batch processes. You cannot take advantage of newer features such as database connectivity, but your integrations will continue to work. Optionally, install the new version alongside the old version and dedicate the old version solely to batch processes.

Generate a new config.properties file from the new GUI

If you originally generated your config.properties file from the graphical user interface, use the new version to set the same properties and generate a new file. Use this new file with the new batch mode interface. For more information, see “Running in Batch Mode” in the online help.

Manually update your config.properties file

If your old config.properties file was created manually, then you must manually update it for the new version. For more information, see [Understanding Installed Directories and Files](#) on page 364.

Using the Command Line Interface

For automated batch operations such as nightly scheduled loads and extractions, run Data Loader from the command line. Before running any batch operation, be sure to include your encrypted password in the configuration file. For more information, see [Encrypting From the Command Line](#) on page 365. From the command line, navigate into the bin directory and type process.bat, which takes the following parameters:

- The directory containing config.properties.
- The name of the batch process bean contained in process-conf.xml.

For more information about using process.bat, see [Running Individual Batch Processes](#) on page 370.

To view tips and instructions, add –help to the command contained in process.bat.

Data Loader runs whatever operation, file, or map is specified in the configuration file that you specify. If you do not specify a configuration directory, the current directory is used. By default, Data Loader configuration files are installed at the following location:

```
C:\Program Files\salesforce.com\Data Loader version number\conf
```

You use the process-conf.xml file to configure batch processing. Set the name of the process in the bean element's id attribute: (for example <bean id="myProcessName">). For more information on batch mode, see “Running in Batch Mode” in the online help.

If you want to implement enhanced logging, use a copy of log-conf.xml.

You can change parameters at runtime by giving *param=value* as program arguments. For example, adding process.operation=insert to the command changes the configuration at runtime.

You can set the minimum and maximum heap size. For example, -Xms256m -Xmx256m sets the heap size to 256 MB.

Note: These topics only apply to Data Loader version 8.0 and later.



Tip:

If you experience login issues in the command line interface after upgrading to a new version of Data Loader, please try re-encrypting your password to solve the problem. For information on the password encryption utility, see [Encrypting From the Command Line](#) on page 365.

Configuring Batch Processes

Use \samples\conf\process-conf.xml to configure your Data Loader processes, which are represented by ProcessRunner beans. A process should have ProcessRunner as the class attribute and the following properties set in the configuration file:

name

Sets the name of the ProcessRunner bean. This value is also used as the non-generic thread name and for configuration backing files (see below).

configOverrideMap

A property of type map where each entry represents a configuration setting: the key is the setting name; the value is the setting value.

enableLastRunOutput

If set to true (the default), output files containing information about the last run, such as `sendMerchandiseFile_lastrun.properties`, are generated and saved to the location specified by `lastRunOutputDirectory`. If set to false, the files are not generated or saved.

lastRunOutputDirectory

The directory location where output files containing information about the last run, such as `sendMerchandiseFile_lastrun.properties`, are written. The default value is `\conf`. If `enableLastRunOutput` is set to false, this value is not used because the files are not generated.

The configuration backing file stores configuration parameter values from the last run for debugging purposes, and is used to load default configuration parameters in `config.properties`. The settings in `configOverrideMap` take precedence over those in the configuration backing file. The configuration backing file is managed programmatically and does not require any manual edits.

For the names and descriptions of available process configuration parameters, see [Data Loader Process Configuration Parameters](#) on page 370.

Data Loader Command Line Operations

When running the Data Loader in batch mode from the command line, several operations are supported. An operation represents the flow of data between Database.com and an external data source such as a CSV file or a database. See the following list of operation names and descriptions.

Extract

Uses a [Salesforce Object Query Language](#) to export a set of records from Database.com, then writes the exported data to a data source. Soft-deleted records are not included.

Extract All

Uses a [Salesforce Object Query Language](#) to export a set of records from Database.com, including both existing and soft-deleted records, then writes the exported data to a data source.

Insert

Loads data from a data source into Database.com as new records.

Update

Loads data from a data source into Database.com, where existing records with matching ID fields are updated.

Upsert

Loads data from a data source into Database.com, where existing records with a matching custom external ID field are updated; records without matches are inserted as new records.

Delete

Loads data from a data source into Database.com, where existing records with matching ID fields are deleted.

Hard Delete

Loads data from a data source into Database.com, where existing records with matching ID fields are deleted without being stored first in the Recycle Bin.

Configuring Database Access

When you run Data Loader in batch mode from the command line, use `\samples\conf\database-conf.xml` to configure database access objects, which you use to extract data directly from a database.

DatabaseConfig Bean

The top-level database configuration object is the `DatabaseConfig` bean, which has the following properties:

sqlConfig

The [SQL configuration bean](#) for the data access object that interacts with a database.

dataSource

The bean that acts as database driver and authenticator. It must refer to an implementation of `javax.sql.DataSource` such as `org.apache.commons.dbcp.BasicDataSource`.

The following code is an example of a `DatabaseConfig` bean:

```
<bean id="MerchandiseInsert"
      class="com.salesforce.dataloader.dao.database.DatabaseConfig"
      singleton="true">
    <property name="sqlConfig" ref="merchandiseInsertSql"/>
</bean>
```

DataSource

The `DataSource` bean sets the physical information needed for database connections. It contains the following properties:

driverClassName

The fully qualified name of the implementation of a JDBC driver.

url

The string for physically connecting to the database.

username

The username for logging in to the database.

password

The password for logging in to the database.

Depending on your implementation, additional information may be required. For example, use `org.apache.commons.dbcp.BasicDataSource` when database connections are pooled.

The following code is an example of a `DataSource` bean:

```
<bean id="oracleRepDataSource"
      class="org.apache.commons.dbcp.BasicDataSource"
      destroy-method="close">
    <property name="driverClassName" value="oracle.jdbc.driver.OracleDriver"/>
```

```
<property name="url" value="jdbc:oracle:thin:@myserver.salesforce.com:1521:TEST"/>
<property name="username" value="test"/>
<property name="password" value="test"/>
</bean>
```

Mapping Columns

When running the Data Loader in batch mode from the command line, you must create a properties file that maps values between Database.com and data access objects.

1. Create a new mapping file and give it an extension of .sdl.
2. Observe the following syntax:

- On each line, pair a data source with its destination.
- In an import file, put the data source on the left, an equals sign (=) as a separator, and the destination on the right. In an export file, put the destination on the left, an equals sign (=) as a separator, and the data source on the right.
- Data sources can be either column names or constants. Surround constants with double quotation marks, as in "sampleconstant". Values without quotation marks are treated as column names.
- Destinations must be column names.
- You may map constants by surrounding them with double quotation marks, as in "United States"=BillingCountry.

3. In your configuration file, use the parameter process.mappingFile to specify the name of your mapping file.

Column Mapping Example for Data Insert

The Database.com fields are on the right.

```
NAME=Name
PRICE=Price__c
DESCRIPTION=Description__c
TOTAL_INVENTORY=Total_Inventory__c
```

Column Mapping Example for Data Export

The Database.com fields are on the left.

```
Name=name
Price__c=price
Description__c=description
```

Column Mapping for Constant Values

Data Loader supports the ability to assign constants to fields when you insert, update, and export data. If you have a field that should contain the same value for each record, you specify that constant in the .sdl mapping file instead of specifying the field and value in the CSV file or the export query.

The constant must be enclosed in double quotation marks. For example, if you're importing data, the syntax is "constantvalue"=field1.

If you have multiple fields that should contain the same value, you must specify the constant and the field names separated by commas. For example, if you're importing data, the syntax would be "constantvalue"=field1, field2.

Here's an example of an .sdl file for inserting data. The Database.com fields are on the right. The first two lines map a data source to a destination field, and the last three lines map a constant to a destination field.

```
Name=Name
Description=Description_c
"12.75"=Price_c
"100"=Total_Inventory_c
"unknown"=Warehouse_location_c
```

A constant must contain at least one alphanumeric character. If you specify a constant value that contains spaces, the spaces are removed when the field data is imported or exported.

Running Individual Batch Processes

To start an individual batch process use \bin\process.bat, which requires the following parameters:

A configuration directory

The default is \conf.

To use an alternate directory, create a new directory and add the following files to it:

- If your process is not interactive, copy process-conf.xml from \samples\conf.
- If your process requires database connectivity, copy database-conf.xml from \samples\conf.
- Copy config.properties from \conf.

A process name

The name of the ProcessRunner bean from \samples\conf\process-conf.xml.

Process Example

```
process ../conf merchandiseMasterProcess
```



Note: You can configure external process launchers such as the Microsoft Windows XP Scheduled Task Wizard to run processes on a schedule.

Data Loader Process Configuration Parameters

When running Data Loader from the command line, you can specify the following configuration parameters in the process-conf.xml file. In some cases, the parameter is also represented in the graphical user interface at **Settings > Settings**.



Tip: A sample process-conf.xml file can be found in the \samples directory that's installed with Data Loader.

Parameter Name	Data Type	Equivalent Option in Settings Dialog	Description
dataAccess.readUTF8	boolean	Read all CSVs with UTF-8 encoding	Select this option to force files to open in UTF-8 encoding, even if they were saved in a different format. Sample value: true
dataAccess.writeUTF8	boolean	Write all CSVs with UTF-8 encoding	Select this option to force files to be written in UTF-8 encoding. Sample value: true
dataAccess.name	string	Not applicable (N/A)	Name of the data source to use, such as a CSV file name. For databases, use the name of the database configuration in <code>database-conf.xml</code> . Sample value: <code>c:\dataloader\data\extractMerchandise.csv</code>
dataAccess.readBatchSize	integer	N/A	Number of records read from the database at a time. The maximum value is 200. Sample value: 50
dataAccess.type	string	N/A	Standard or custom data source type. Standard types are <code>csvWriter</code> , <code>csvRead</code> , <code>databaseWrite</code> , and <code>databaseRead</code> . Sample value: <code>csvWrite</code>
dataAccess.writeBatchSize	integer	N/A	Number of records written to the database at a time. The maximum value is 2,000. Note the implication for a large parameter value: if an error occurs, all records in the batch are rolled back. In contrast, if the value is set to 1, each record is processed individually (not in batch) and errors are specific to a given record. We recommend setting the value to 1 when you need to diagnose problems with writing to a database. Sample value: 500
process.enableExtractSuccessOutput	boolean	Generate status files for exports	Select this option to generate success and error files when exporting data. Sample value: true
process.enableLastRunOutput	boolean	N/A	When running Data Loader in batch mode, you can disable the generation of output files such as <code>sendMerchandiseFile_lastRun.properties</code> .

Parameter Name	Data Type	Equivalent Option in Settings Dialog	Description
process.encryptionKeyFile	string (file name)	N/A	<p>Files of this type are saved by default to the <code>conf</code> directory. To stop the writing of these files, set this option to <code>false</code>.</p> <p>Alternatively, you can change the location of the directory where these files are saved, using <code>process.lastRunOutputDirectory</code>.</p> <p>Sample value: <code>true</code></p>
process.initialLastRunDate	date	N/A	<p>Name of the file that contains the encryption key. See Encrypting From the Command Line on page 365.</p> <p>Sample value: <code>c:\dataloader\conf\my.key</code></p>
process.lastRunOutputDirectory	string (directory)	N/A	<p>The initial setting for the <code>process.lastRunDate</code> parameter, which can be used in a SQL string and is automatically updated when a process has run successfully. For an explanation of the date format syntax, see Date Formats on page 348.</p> <p>Format must be <code>yyyy-MM-ddTHH:mm:ss.SSS+-HHmm</code>. For example: <code>2006-04-13T13:50:32.423-0700</code></p> <p>When running Data Loader in batch mode, you can change the location where output files such as <code>sendMerchandiseFile_lastRun.properties</code> are written. Files of this type are saved by default to the <code>\conf</code> directory. To change the location, change the value of this option to the full path where the output files should be written.</p> <p>Alternatively, you can stop the files from being written, using <code>process.enableLastRunOutput</code>.</p>
process.loadRowToStartAt	number	Start at row	<p>If your last operation failed, you can use this setting to begin where the last successful operation finished.</p> <p>Sample value: <code>1008</code></p>

Parameter Name	Data Type	Equivalent Option in Settings Dialog	Description
process.mappingFile	string (file name)	N/A	Name of the field mapping file to use. See Mapping Columns on page 369. Sample value: c:\dataloader\conf\merchandiseExtractMap.sdl
process.operation	string	N/A	The operation to perform. See Data Loader Command Line Operations on page 367. Sample value: extract
process.statusOutputDirectory	string (directory)	N/A	The directory where “success” and “error” output files are saved. The file names are automatically generated for each operation unless you specify otherwise in process-conf.xml. Sample value: c:\dataloader\status
process.outputError	string (file name)	N/A	The name of the CSV file that stores error data from the last operation. Sample value: c:\dataloader\status\myProcessErrors.csv
process.outputSuccess	string (file name)	N/A	The name of the CSV file that stores success data from the last operation. See also process.enableExtractSuccessOutput on page 371. Sample value: c:\dataloader\status\myProcessSuccesses.csv
process.useEuropeanDates	boolean	Use European date format	Select this option to support the date formats dd/MM/yyyy and dd/MM/yyyy HH:mm:ss. Sample value: true
sfdc.bulkApiCheckStatusInterval	integer	N/A	The number of milliseconds to wait between successive checks to determine if the asynchronous Bulk API operation is complete or how many records have been processed. See also sfdc.useBulkApi . We recommend a value of 5000. Sample value: 5000
sfdc.bulkApiSerialMode	boolean	Enable serial	Select this option to use serial instead of parallel processing for Bulk API. Processing in parallel can cause database contention. When this is

Parameter Name	Data Type	Equivalent Option in Settings Dialog	Description
		mode for Bulk API	severe, the load may fail. Using serial mode guarantees that batches are processed one at a time. Note that using this option may significantly increase the processing time for a load. See also sfdc.useBulkApi .
			Sample value: false
sfdc.connectionTimeoutSecs	integer	N/A	The number of seconds to wait for a connection during API calls. Sample value: 60
sfdc.debugMessages	boolean	N/A	If true, enables SOAP message debugging. By default, messages are sent to STDOUT unless you specify an alternate location in sfdc.debugMessagesFile . Sample value: false
sfdc.debugMessagesFile	string (file name)	N/A	See process.enableExtractSuccessOutput on page 371. Stores SOAP messages sent to or from Database.com. As messages are sent or received, they are appended to the end of the file. As the file does not have a size limit, please monitor your available disk storage appropriately. Sample value: \lexiloader\status\sfdcSoapTrace.log
sfdc.enableRetries	boolean	N/A	If true, enables repeated attempts to connect to Database.com servers. See sfdc.maxRetries on page 375 and sfdc.minRetrySleepSecs on page 375. Sample value: true
sfdc.endpoint	URL	Server host	Enter the URL of the Database.com server with which you want to communicate. For example, if you are loading data into a test database , change the URL to https://test.database.com . Sample production value: https://login.database.com/services/Soap/u/24.0
sfdc.entity	string	N/A	The Database.com object used in the operation. Sample value: Merchandise__c

Parameter Name	Data Type	Equivalent Option in Settings Dialog	Description
sfdc.externalIdField	string	N/A	<p>Used in upsert operations; specifies the custom field with the “External ID” attribute that is used as a unique identifier for data matching. For more information about external IDs, see Custom Field Attributes on page 101.</p> <p>Sample value: LegacySKU__c</p>
sfdc.extractionRequestSize	integer	Query request size	<p>In a single export or query operation, records are returned from Database.com in increments of this size. The maximum value is 2,000 records. Larger values may improve performance but use more memory on the client.</p> <p>Sample value: 500</p>
sfdc.extractionSOQL	string	N/A	<p>The SOQL query for the data export. For more information on SOQL, see the Web Services API Developer's Guide.</p> <p>Sample value: SELECT Name, Description__c, Price__c, Total_Inventory__c FROM Merchandise__c</p>
sfdc.insertNulls	boolean	Insert null values	<p>Select this option to insert blank mapped values as null values during data operations. Note that when you are updating records, this option instructs Data Loader to overwrite any existing data in mapped fields.</p> <p>Sample value: false</p>
sfdc.loadBatchSize	integer	Batch size	<p>In a single insert, update, upsert, or delete operation, records moving to or from Database.com are processed in increments of this size. The maximum value is 200. We recommend a value between 50 and 100.</p> <p>Sample value: 100</p>
sfdc.maxRetries	integer	N/A	<p>The maximum number of repeated attempts to connect to Database.com. See sfdc.enableRetries on page 374.</p> <p>Sample value: 3</p>
sfdc.minRetrySleepSecs	integer	N/A	<p>The minimum number of seconds to wait between connection retries. The wait time</p>

Parameter Name	Data Type	Equivalent Option in Settings Dialog	Description
<code>sfdc.enableRetries</code>	integer	Retries	increases with each try. See sfdc.enableRetries on page 374. Sample value: 2
<code>sfdc.noCompression</code>	boolean	Compression	Compression enhances the performance of Data Loader and is turned on by default. You may want to disable compression if you need to debug the underlying SOAP messages. To turn off compression, enable this option. Sample value: false
<code>sfdc.password</code>	encrypted string	N/A	An encrypted Database.com password that corresponds to the username provided in sfdc.username . See also Encrypting From the Command Line on page 365. Sample value: 4285b36161c65a22
<code>sfdc.proxyHost</code>	URL	Proxy host	The host name of the proxy server, if applicable. Sample value: http://myproxy.internal.company.com
<code>sfdc.proxyPassword</code>	encrypted string	Proxy password	An encrypted password that corresponds to the proxy username provided in sfdc.proxyUsername . See also Encrypting From the Command Line on page 365. Sample value: 4285b36161c65a22
<code>sfdc.proxyPort</code>	integer	Proxy port	The proxy server port. Sample value: 8000
<code>sfdc.proxyUsername</code>	string	Proxy username	The username for proxy server authentication. Sample value: jane.doe
<code>sfdc.resetUrlOnLogin</code>	boolean	Reset URL on Login	By default, Database.com resets the URL after login to the one specified in sfdc.endpoint . To turn off this automatic reset, disable this option by setting it to <code>false</code> . Valid values: <code>true</code> (default), <code>false</code>

Parameter Name	Data Type	Equivalent Option in Settings Dialog	Description
sfdc.timeoutSecs	integer	Timeout	<p>Specify how many seconds Data Loader waits to receive a response back from the server before returning an error for the request.</p> <p>Sample value: 540</p>
sfdc.timezone	string	Time Zone	<p>If a date value does not include a time zone, this value is used.</p> <ul style="list-style-type: none"> • If no value is specified, the time zone of the computer where Data Loader is installed is used. • If an incorrect value is entered, GMT is used as the time zone and this fact is noted in the Data Loader log. <p>Valid values are any time zone identifier which can be passed to the Java <code>getTimeZone(java.lang.String)</code> method. The value can be a full name such as <code>America/Los_Angeles</code>, or a custom ID such as <code>GMT-8:00</code>.</p> <p>You can retrieve the default value by running the <code>TimeZone.getDefault()</code> method in Java. This value is the time zone on the computer where Data Loader is installed.</p>
sfdc.truncateFields	boolean	Allow field truncation	<p>Select this option to truncate data in the following types of fields when loading that data into Database.com: Email, Multi-select Picklist, Phone, Picklist, Text, and Text (Encrypted).</p> <p>In Data Loader versions 14.0 and earlier, values for fields of those types are truncated by Data Loader if they are too large. In Data Loader version 15.0 and later, the load operation fails if a value is specified that is too large.</p> <p>Selecting this option allows you to specify that the previous behavior, truncation, be used instead of the new behavior in Data Loader versions 15.0 and later. This option is selected by default and has no effect in versions 14.0 and earlier.</p> <p>This option is not available if the Use Bulk API option is selected. In that case, the load</p>

Parameter Name	Data Type	Equivalent Option in Settings Dialog	Description
<code>sfdc.useBulkApi</code>	boolean	Use Bulk API	operation fails for the row if a value is specified that is too large for the field. Sample value: true
<code>sfdc.username</code>	string	N/A	Select this option to use the Bulk API to insert, update, upsert, delete, and hard delete records. The Bulk API is optimized to load or delete a large number of records asynchronously. It's faster than the default SOAP-based API due to parallel processing and fewer network round-trips. See also sfdc.bulkApiSerialMode . Sample value: true
			Database.com username. See sfdc.password . Sample value: jdoe@mycompany.com

SQL Configuration

When running the Data Loader in batch mode from the command line, the `SqlConfig` class contains configuration parameters for accessing specific data in the database. As shown in the code samples below, queries and inserts are different but very similar. The bean must be of type `com.salesforce.dataloader.dao.database.SqlConfig` and have the following properties:

`sqlString`

The SQL code to be used by the data access object.

The SQL can contain replacement parameters that make the string dependent on configuration or operation variables. Replacement parameters must be delimited on both sides by "@" characters. For example, @process.lastRunDate@.

`sqlParams`

A property of type `map` that contains descriptions of the replacement parameters specified in `sqlString`. Each entry represents one replacement parameter: the key is the replacement parameter's name, the value is the fully qualified Java type to be used when the parameter is set on the SQL statement. Note that "java.sql" types are sometimes required, such as `java.sql.Date` instead of `java.util.Date`. For more information, see [the official JDBC API documentation](#).

`columnNames`

Used when queries (SELECT statements) return a JDBC `ResultSet`. Contains column names for the data outputted by executing the SQL. The column names are used to access and return the output to the caller of the `DataReader` interface.

SQL Query Bean Example

```
<bean id="merchandiseQuerySql"
  class="com.salesforce.dataloader.dao.database.SqlConfig"
  singleton="true">
<property name="sqlString">
  <value>
    SELECT distinct
      merchandise_name,
      merchandise_description,
      merchandise_price
    from
      EXTDB.MERCHANDISE merch,
    where
      merchandise_last_update_date > @process.lastRunDate@
  </value>
</property>
<property name="columNames">
  <list>
    <value>name</value>
    <value>description</value>
    <value>price</value>
  </list>
</property>
<property name="sqlParams">
  <map>
    <entry key="process.lastRunDate" value="java.sql.Date"/>
  </map>
</property>
</bean>
```

SQL Insert Bean Example

```
<bean id="merchandiseInsertSql"
  class="com.salesforce.dataloader.dao.database.SqlConfig"
  singleton="true">
<property name="sqlString">
  <value>
    INSERT INTO EXTDB.MERCH (
      DESCRIPTIONS, EXPIRATION_DATES)
    VALUES (@desc@, @expire@)
  </value>
</property>
<property name="sqlParams">
  <map>
    <entry key="desc" value="java.lang.String"/>
    <entry key="expire" value="java.sql.Date"/>
  </map>
</property>
</bean>
```

Managing Application Logic

Working with Apex

Understanding Apex

Apex Code Overview

Apex is a strongly typed, object-oriented programming language that allows developers to execute flow and transaction control statements on Database.com in conjunction with calls to the Force.com API. Using syntax that looks like Java and acts like database stored procedures, Apex enables developers to add business logic to most system events. Apex code can be initiated by Web service requests and from triggers on objects.

Apex can be stored on the platform in two different forms:

- A *class* is a template or blueprint from which Apex objects are created. Classes consist of other classes, user-defined methods, variables, exception types, and static initialization code under **Develop > Apex Classes**. See [Managing Apex Classes](#) on page 381.
- A *trigger* is Apex code that executes before or after specific data manipulation language (DML) events occur, such as before object records are inserted into the database, or after records have been deleted. Triggers are stored as metadata in Database.com. A list of all triggers in your organization is located at **Develop > Apex Triggers**. See [Managing Apex Triggers](#) on page 386.

Apex generally runs in system context; that is, the current user's permissions, field-level security, and sharing rules aren't taken into account during code execution.

You must have at least 75% of your Apex covered by unit tests before you can deploy your code to production environments. In addition, all triggers must have some test coverage. See [About Apex Unit Tests](#) on page 403.

After creating your classes and triggers, as well as your tests, replay the execution using the [Developer Console](#).



Note: You can add, edit, or delete Apex using the Database.com user interface only in a [test database](#) organization. In a Database.com production organization, you can only make changes to Apex by using the Metadata API `deploy` call, the Force.com IDE, or the Force.com Migration Tool. The Force.com IDE and Force.com Migration Tool are free resources provided by salesforce.com to support its users and partners, but are not considered part of our Services for purposes of the salesforce.com Master Subscription Agreement.

For more information on the syntax and use of Apex, see the [Database.com Apex Code Developer's Guide](#).

Development Overview

Click **Develop** to access the following options. These app builder tools may require some programming knowledge:

Apex Classes

Define Apex classes that you can use to add additional business logic to your custom applications.

Apex Triggers

View all Apex triggers defined for your organization.

Working with Apex Test Execution

Run Apex unit tests and view test results.

API

Download WSDL files that allow you to integrate external applications with Database.com.

Custom Settings

Create and manage custom data for your organization.

Remote Access

Create and manage a remote access application. A remote access application is an application external to Database.com that uses the OAuth protocol to verify both the Database.com user and the external application.

Tools

Download tools that can assist you with building, debugging, testing, and deploying Apex.

Managing Apex Classes

Managing Apex Classes

User Permissions Needed

To define, edit, delete, set security, set version settings, show dependencies, and run tests “Author Apex” for Apex classes:

An Apex class is a template or blueprint from which Apex objects are created. Classes consist of other classes, user-defined methods, variables, exception types, and static initialization code. Once successfully saved, class methods or variables can be invoked by other Apex code, or through the Web services API (or AJAX Toolkit) for methods that have been designated with the `webService` keyword.

The Apex Classes page enables you to create and manage Apex classes. To access the Apex Classes page, click **Develop > Apex Classes**.

To create an Apex class, from the Apex Classes page, click **New** and write your Apex code in the editor. See [Defining Apex Classes](#) on page 382.

While developers can write class methods according to the syntax outlined in the [Database.com Apex Code Developer's Guide](#), classes can also be automatically generated by consuming a WSDL document that is stored on a local hard drive or network. Creating a class by consuming a WSDL document allows developers to make callouts to the external Web service in their Apex. From the Apex Classes page, click **Generate From WSDL** to generate an Apex class from a WSDL document.

Once you have created an Apex class, you can do any of the following:

- Click **Edit** next to the class name to modify its contents in a simple editor.
- Click **Del** next to the class name to delete the class from your organization.

Note:



◊ You can add, edit, or delete Apex using the Database.com user interface only in a [test database](#) organization. In a Database.com production organization, you can only make changes to Apex by using the Metadata API `deploy` call, the Force.com IDE, or the Force.com Migration Tool. The Force.com IDE and Force.com Migration Tool are free resources provided by salesforce.com to support its users and partners, but are not considered part of our Services for purposes of the salesforce.com Master Subscription Agreement.

- If an Apex class has any methods defined as a `webService`, you can click **WSDL** next to the class name to generate a WSDL document from the class contents. This document contains all of the information necessary for a client to consume Apex Web service methods. All class methods with the `webService` keyword are included in the resulting WSDL document.
- Click **Security** next to the class name to select the [profiles](#) that are allowed to execute methods in the class from top-level entry points, such as Web service methods.
- Click **Calculate your organization's code coverage** to find out how much of the Apex code in your organization is currently covered by unit tests. This percentage is based on the latest test results. If you have no test results, code coverage is listed at 100%. For more information, see [About Apex Unit Tests](#) on page 403.
- Click the percentage number in the Code Coverage column to see which lines in a class have been covered by Apex unit tests.

- If you have unit tests in at least one Apex class, click **Run All Tests** to run all the unit tests in your organization. For more information, see [About Apex Unit Tests](#) on page 403.
- Click **Compile all classes** to compile all the Apex classes in your organization.

Defining Apex Classes

User Permissions Needed

To define, edit, delete, set security, set version settings, show dependencies, and run tests “Author Apex” for Apex classes:

[Apex classes](#) are stored as metadata in Database.com.



Note: You can add, edit, or delete Apex using the Database.com user interface only in a [test database](#) organization. In a Database.com production organization, you can only make changes to Apex by using the Metadata API `deploy` call, the Force.com IDE, or the Force.com Migration Tool. The Force.com IDE and Force.com Migration Tool are free resources provided by salesforce.com to support its users and partners, but are not considered part of our Services for purposes of the salesforce.com Master Subscription Agreement.

To create a class:

1. Click **Develop > Apex Classes**.
2. Click **New**.
3. Click **Version Settings** to specify the version of Apex and the API used with this class. Use the default values for all versions. This associates the class with the most recent version of Apex and the API. You can specify an older version of Apex and the API to maintain specific behavior.
4. In the class editor, enter the Apex code for the class. A single class can be up to 1 million characters in length, not including comments, test methods, or classes defined using `@isTest`.
5. Click **Save** to save your changes and return to the class detail screen, or click **Quick Save** to save your changes and continue editing your class. Your Apex class must compile correctly before you can save your class.

Once saved, classes can be invoked through class methods or variables by other Apex code, such as a trigger.



Note: To aid backwards-compatibility, classes are stored with the version settings for a specified version of Apex and the API. Additionally, classes are stored with an `isValid` flag that is set to `true` as long as dependent metadata has not changed since the class was last compiled. If any changes are made to object names or fields that are used in the class, including superficial changes such as edits to an object or field description, or if changes are made to a class that calls this class, the `isValid` flag is set to `false`. When a trigger or Web service call invokes the class, the code is recompiled and the user is notified if there are any errors. If there are no errors, the `isValid` flag is reset to `true`.

Creating an Apex Class from a WSDL

User Permissions Needed

To define, edit, delete, set security, set version settings, show dependencies, and run tests “Author Apex” for Apex classes:

An [Apex class](#) can be automatically generated from a WSDL document that is stored on a local hard drive or network. Creating a class by consuming a WSDL document allows developers to make callouts to the external Web service in their Apex.

To access this functionality:

1. In the application, click **Develop > Apex Classes**.
2. Click **Generate from WSDL**.
3. Click **Browse** to navigate to a WSDL document on your local hard drive or network, or type in the full path. This WSDL document is the basis for the Apex class you are creating.



Note:

The WSDL document that you specify might contain a SOAP endpoint location that references an outbound port.

For security reasons, Database.com restricts the outbound ports you may specify to one of the following:

- 80: This port only accepts HTTP connections.
- 443: This port only accepts HTTPS connections.
- 1024–66535 (inclusive): These ports accept HTTP or HTTPS connections.

4. Click **Parse WSDL** to verify the WSDL document contents. The application generates a default class name for each namespace in the WSDL document and reports any errors. Parsing will fail if the WSDL contains schema types or schema constructs that are not supported by Apex classes, or if the resulting classes exceed 1 million character limit on Apex classes. For example, the Database.com SOAP API WSDL cannot be parsed.
5. Modify the class names as desired. While you can save more than one WSDL namespace into a single class by using the same class name for each namespace, Apex classes can be no more than 1 million characters total.
6. Click **Generate Apex**. The final page of the wizard shows which classes were successfully generated, along with any errors from other classes. The page also provides a link to view successfully generated code.

The successfully-generated Apex class includes stub and type classes for calling the third-party Web service represented by the WSDL document. These classes allow you to call the external Web service from Apex. For an example, see the *Database.com Apex Code Developer's Guide*.

Note the following about the generated Apex:

- If a WSDL document contains an Apex reserved word, the word is appended with `_x` when the Apex class is generated. For example, `limit` in a WSDL document converts to `limit_x` in the generated Apex class. For a list of reserved words, see the *Database.com Apex Code Developer's Guide*.
- If an operation in the WSDL has an output message with more than one element, the generated Apex wraps the elements in an inner class. The Apex method that represents the WSDL operation returns the inner class instead of the individual elements.

Viewing Apex Classes

User Permissions Needed
To define, edit, delete, set security, set version settings, show dependencies, and run tests “Author Apex” for Apex classes:

After you have created a class, you can view the code contained in the class, as well as the API against which the class was saved, and whether the class is valid or active. Click **Develop > Apex Classes**, then click the name of the class you want to view. While viewing a class, you can do any of the following:

- Click **Edit** to make changes to the class.



Note: You can add, edit, or delete Apex using the Database.com user interface only in a [test database organization](#). In a Database.com production organization, you can only make changes to Apex by using the Metadata API `deploy` call, the Force.com IDE, or the Force.com Migration Tool. The Force.com IDE and Force.com Migration Tool are free resources provided by salesforce.com to support its users and partners, but are not considered part of our Services for purposes of the salesforce.com Master Subscription Agreement.

- Click **Delete** to delete the class.
- If your class has a method defined as a `webservice`, click **Generate WSDL** to generate a WSDL document based on the class.



Note: You cannot generate a WSDL document for classes defined as `isTest`.

- Click **Download** to download a copy of your Apex.
- Click **Run Test** to run the [unit tests](#) contained in the class.
- Click **Security** to set the [Apex class level security](#).
- Click **Show Dependencies** to display the items, such as fields, objects, or other classes, that must exist for this class to be valid.

The **Class Summary** tab displays the prototype of the class; that is, the classes, methods and variables that are available to other Apex code. The **Class Summary** tab lists the access level and signature for each method and variable in an Apex class, as well as any inner classes. If there is no prototype available, this tab is not available.

The **Log Filters** tab displays the debug log categories and debug log levels that you can set for the class. For more information, see [Debug Log Filtering for Apex Classes and Apex Triggers](#) on page 401.

Setting Apex Class Access

Apex Class Security Overview

You can specify which users can execute methods in a particular top-level Apex class based on their profile or an associated permission set. These permissions only apply to Apex class methods, such as Web service methods. Triggers always fire on trigger events (such as `insert` or `update`), regardless of a user's permissions.

Permission for an Apex class is checked at the top level only. For example, if class A calls class B, and a user profile has access only to class A but not class B, the user can still execute the code in class A.

You can set Apex class security via:

- [The Apex class list page](#)
- [An Apex class detail page](#)
- [Permission sets](#)
- [Profiles](#)

Setting Apex Class Access from the Class List Page

User Permissions Needed	
To set Apex class security:	“Author Apex”
	AND
	“Customize Application”

1. Click **Develop > Apex Classes**.
2. Next to the name of the class that you want to restrict, click **Security**.
3. Select the profiles that you want to enable from the Available Profiles list and click **Add**, or select the profiles that you want to disable from the Enabled Profiles list and click **Remove**.
4. Click **Save**.

Setting Apex Class Access from the Class Detail Page

User Permissions Needed	
To set Apex class security:	“Author Apex”
	AND
	“Customize Application”

1. Click **Develop > Apex Classes**.
2. Click the name of the class that you want to restrict.
3. Click **Security**.
4. Select the profiles that you want to enable from the Available Profiles list and click **Add**, or select the profiles that you want to disable from the Enabled Profiles list and click **Remove**.
5. Click **Save**.

Setting Apex Class Access from Permission Sets

User Permissions Needed	
To edit Apex class access settings:	“Manage Users”

You can specify which methods in a top-level Apex class are executable for a permission set. These settings only apply to Apex class methods, such as Web service methods. Triggers always fire on trigger events (such as `insert` or `update`), regardless of permission settings.

1. Click **Manage Users > Permission Sets**.
2. Select a permission set.
3. Click **Apex Class Access**.
4. Click **Edit**.
5. Select the Apex classes that you want to enable from the Available Apex Classes list and click **Add**, or select the Apex classes that you want to disable from the Enabled Apex Classes list and click **Remove**.
6. Click **Save**.

Setting Apex Class Access from Profiles

User Permissions Needed	
To edit profiles:	“Manage Users” AND “Customize Application”

You can specify which methods in a top-level Apex class are executable for a profile. These settings only apply to Apex class methods, such as Web service methods. Triggers always fire on trigger events (such as `insert` or `update`), regardless of profile settings.

1. Click **Manage Users > Profiles**.
2. Select a profile.
3. In the Apex Class Access page or related list, click **Edit**.
4. Select the Apex classes that you want to enable from the Available Apex Classes list and click **Add**, or select the Apex classes that you want to disable from the Enabled Apex Classes list and click **Remove**.
5. Click **Save**.

Managing Version Settings for Apex

User Permissions Needed
To define, edit, delete, set security, set version settings, show dependencies, and run tests “Author Apex” for Apex classes:

To aid backwards-compatibility, classes are stored with the version settings for a specified version of Apex and the API. This ensures that as Apex and the API evolve in subsequent released versions, a class or trigger is still bound to versions with specific, known behavior.

To set the Database.com API and Apex version for a class or trigger:

1. Edit either a class or trigger, and click **Version Settings**.
2. Select the version of the Database.com API. This is also the version of Apex associated with the class or trigger.
3. Click **Save**.

Managing Apex Triggers

See Also:

[Managing Version Settings for Apex](#)

Managing Apex Triggers

User Permissions Needed

To define, edit, delete, and show dependencies for Apex triggers:

“Author Apex”

A trigger is Apex code that executes before or after specific data manipulation language (DML) events occur, such as before object records are inserted into the database, or after records have been deleted.

Triggers are stored as metadata in Database.com. A list of all triggers in your organization is located at **Develop > Apex Triggers**. In addition to this list, triggers are associated and stored with specific objects. For custom objects, triggers are located on the object detail page at **Create > Objects > Object Name**.

Click **Create > Objects > Object Name**. In the Triggers section, click **New** to create an Apex trigger.



Note: You can only create triggers from the associated object, not from the Apex Triggers page.

Once you have created an Apex trigger:

- Click **Edit** next to the trigger name to modify its contents in a simple editor.
- Click **Del** next to the trigger name to delete the trigger from your organization.
- Click the percentage number in the Code Coverage column to see which lines in a trigger have been covered by Apex unit tests.



Note: You can add, edit, or delete Apex using the Database.com user interface only in a [test database](#) organization. In a Database.com production organization, you can only make changes to Apex by using the Metadata API `deploy` call, the Force.com IDE, or the Force.com Migration Tool. The Force.com IDE and Force.com Migration Tool are free resources provided by salesforce.com to support its users and partners, but are not considered part of our Services for purposes of the salesforce.com Master Subscription Agreement.

Defining Apex Triggers

User Permissions Needed

To define Apex triggers:

“Author Apex”

Apex triggers are stored as metadata in the application under the object with which they are associated.



Note: You can add, edit, or delete Apex using the Database.com user interface only in a [test database](#) organization. In a Database.com production organization, you can only make changes to Apex by using the Metadata API `deploy` call, the Force.com IDE, or the Force.com Migration Tool. The Force.com IDE and Force.com Migration Tool are free resources provided by salesforce.com to support its users and partners, but are not considered part of our Services for purposes of the salesforce.com Master Subscription Agreement.

To define a trigger:

1. For a custom object, click **Create > Objects** and click the name of the object.
2. In the Triggers related list, click **New**.
3. Click **Version Settings** to specify the version of Apex and the API used with this trigger.
4. Select the **Is Active** checkbox if the trigger should be compiled and enabled. Leave this checkbox deselected if you only want to store the code in your organization's metadata. This checkbox is selected by default.

5. In the Body text box, enter the Apex for the trigger. A single trigger can be up to 1 million characters in length.

To define a trigger, use the following syntax:

```
trigger triggerName on ObjectName (trigger_events) {
    code_block
}
```

where *trigger_events* can be a comma-separated list of one or more of the following events:

- before insert
- before update
- before delete
- after insert
- after update
- after delete
- after undelete



Note:

- You can only use the `webService` keyword in a trigger when it is in a method defined as asynchronous; that is, when the method is defined with the `@future` keyword.
- A trigger invoked by an `insert`, `delete`, or `update` of a recurring event or recurring task results in a runtime error when the trigger is called in bulk from the Force.com API.

6. Click **Save**.



Note: Triggers are stored with an `isValid` flag that is set to `true` as long as dependent metadata has not changed since the trigger was last compiled. If any changes are made to object names or fields that are used in the trigger, including superficial changes such as edits to an object or field description, the `isValid` flag is set to `false` until the Apex compiler reprocesses the code. Recompiling occurs when the trigger is next executed, or when a user re-saves the trigger in metadata.

Viewing Apex Trigger Details

User Permissions Needed	
To view Apex triggers:	“Author Apex”

Apex triggers are stored as metadata in the application under the object with which they are associated. You can also view all triggers by clicking **Develop > Apex Triggers**.

To view the details for a trigger, click **Develop > Apex Triggers**, then click the name of the trigger. You can also access the trigger details from the object. For a custom object, click **Create > Objects**, click the name of the object, then click the name of the trigger.

From the trigger detail page, you can do any of the following:

- Click **Edit** to modify the contents of the trigger.
- Click **Delete** to delete the trigger from your organization.

- Click **Show Dependencies** to display the items, such as fields, or classes, that are referenced by the Apex code contained in the trigger.
- Click **Download Apex** to download the text of the trigger. The file is saved with the name of the trigger as the file name, with the filetype of .trg.

The trigger detail page shows the following information for a trigger:

- The name of the trigger
- The name of the object with which the trigger is associated.
- The API version that the trigger has been saved against.
- Whether a trigger is valid.



Note: Triggers are stored with an `isValid` flag that is set to `true` as long as dependent metadata has not changed since the trigger was last compiled. If any changes are made to object names or fields that are used in the trigger, including superficial changes such as edits to an object or field description, the `isValid` flag is set to `false` until the Apex compiler reprocesses the code. Recompiling occurs when the trigger is next executed, or when a user re-saves the trigger in metadata.

- Whether the trigger is active.
- The text of the Apex code contained in the trigger.

The **Log Filters** tab displays the debug log categories and debug log levels that you can set for the trigger. For more information, see [Debug Log Filtering for Apex Classes and Apex Triggers](#) on page 401.

Debugging Apex

Debugging Apex Using Debug Logs

Understanding Debug Logs

What is a Debug Log?

User Permissions Needed	
To use the Developer Console:	“View All Data”
To use the execute anonymous text entry box:	“Author Apex”
To save changes to Apex classes and triggers:	“Author Apex”

A *debug log* records database operations, system processes, and errors that occur when executing a transaction or while running unit tests. The system generates a debug log for a user every time that user executes a transaction that is included in the filter criteria.

Transactions can be generated from the following:

- API
- executeanonymous calls
- Web services

The [filter criteria](#) set for the user, the Developer Console or the API header determines what is included in the debug log. Debug logs can contain information about:

- Database changes
- HTTP callouts

- Apex errors
- Resources used by Apex
- Automated workflow processes, such as:
 - ◊ Workflow rules
 - ◊ Validation rules

The following are examples of when you use a debug log:

- You're a developer creating a custom application and use the debug log to validate some of the application's behavior, such as if the application made callouts to an external system. You could set the debug log filter to check for callouts, then in the debug log, view information about the success and duration of those callouts.
- You're an administrator for an organization, and a user reports having difficulties. You could start to monitor the debug logs for that user, have them step through the transaction, and then use the debug log to view the system details of the transaction.

Debug Log Limits

The following are the limits for debug logs:

- Once a user is added, that user can record up to 20 debug logs. After a user reaches this limit, debug logs stop being recorded for that user. Click **Reset** on the Monitoring Debug logs page to reset the number of logs for that user back to 20. Any existing logs are not overwritten.
- Each debug log can only be 2 MB. Debug logs that are larger than 2 MB in size are truncated.
- Each organization can retain up to 50 MB of debug logs. Once your organization has reached 50 MB of debug logs, the oldest debug logs start being overwritten.

Debug Log Truncation

Debug logs are truncated starting from the oldest log entries. The newer log entries are preserved. This allows you to have access to the most pertinent information leading to an error that you are diagnosing. The debug log gets truncated by 200 KBytes when it reaches its maximum size of 2 MB. Some log entries don't get truncated and will always be part of the debug log, even if they're part of the oldest log entries, because they're necessary for processing the debug log. However, other log information that appears between the start and end lines of these log entries will be removed as part of log truncation. The following are the events that are associated with non-deletable log entries.

- EXECUTION_STARTED
- EXECUTION_FINISHED
- CODE_UNIT_STARTED
- CODE_UNIT_FINISHED
- METHOD_ENTRY
- METHOD_EXIT
- CONSTRUCTOR_ENTRY
- CONSTRUCTOR_EXIT
- SOQL_EXECUTE_BEGIN
- SOQL_EXECUTE_END
- SOSL_EXECUTE_BEGIN
- SOSL_EXECUTE_END
- CALLOUT_REQUEST
- CALLOUT_RESPONSE
- FATAL_ERROR

Debug Log Details

User Permissions Needed	
To use the Developer Console:	“View All Data”
To use the execute anonymous text entry box:	“Author Apex”
To save changes to Apex classes and triggers:	“Author Apex”

Inspecting the Debug Log Sections

After you generate a debug log, the type and amount of information listed depends on the [filter values](#) you set for the user. However, the format for a debug log is always the same.

A debug log has the following sections:

Header

The header contains the following information:

- The version of the API used during the transaction.
- The [log category and level](#) used to generate the log. For example:

The following is an example of a header:

```
22.0
APEX_CODE,DEBUG;APEX_PROFILING,INFO;CALLOUT,INFO;DB,INFO;SYSTEM,DEBUG;VALIDATION,INFO;VISUALFORCE,INFO;
WORKFLOW,INFO
```

In this example, the API version is 22.0, and the following debug log categories and levels have been set:

Apex Code	DEBUG
Apex Profiling	INFO
Callout	INFO
Database	INFO
System	DEBUG
Validation	INFO
Visualforce	INFO
Workflow	INFO

Execution Units

An execution unit is equivalent to a transaction. It contains everything that occurred within the transaction. The execution is delimited by `EXECUTION_STARTED` and `EXECUTION_FINISHED`.

Code Units

A code unit is a discrete unit of work within a transaction. For example, a trigger is one unit of code, as is a `webService` method, or a validation rule.



Note: A class is **not** a discrete unit of code.

Units of code are indicated by CODE_UNIT_STARTED and CODE_UNIT_FINISHED. Units of work can embed other units of work. For example:

```
EXECUTION_STARTED
CODE_UNIT_STARTED|[EXTERNAL]execute_anonymous_apex
CODE_UNIT_STARTED|[EXTERNAL]MyTrigger on Merchandise trigger event BeforeInsert for
[new]
CODE_UNIT_FINISHED <-- The trigger ends
CODE_UNIT_FINISHED <-- The executeAnonymous ends
EXECUTION_FINISHED
```

Units of code include, but are not limited to, the following:

- Triggers
- Workflow invocations and time-based workflow
- Validation rules
- @future method invocations
- Web service invocations
- executeAnonymous calls
- Execution of the batch Apex start and finish methods, as well as each execution of the execute method
- Execution of the Apex System.Schedule execute method

Log Lines

Included inside the units of code. These indicate what code or rules are being executed, or messages being specifically written to the debug log. For example:

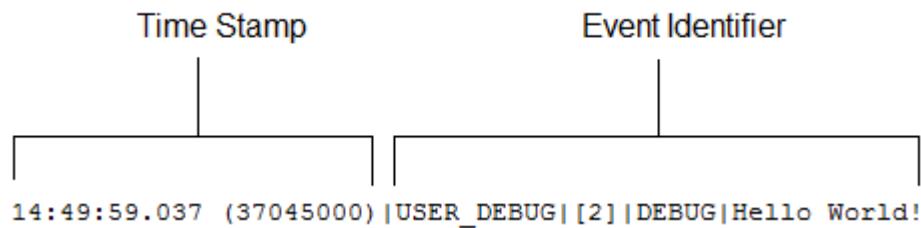


Figure 2: Debug Log Line Example

Log lines are made up of a set of fields, delimited by a pipe (|). The format is:

- *timestamp*: consists of the time when the event occurred and a value between parentheses. The time is in the user's time zone and in the format *HH:mm:ss.sss*. The value represents the time elapsed in nanoseconds since the start of the request. The elapsed time value is excluded from logs reviewed in the Developer Console.
- *event identifier*: consists of the specific event that triggered the debug log being written to, such as SAVEPOINT_RESET or VALIDATION_RULE, and any additional information logged with that event, such as the method name or the line and character number where the code was executed.

Additional Log Data

In addition, the log contains the following information:

- Cumulative resource usage—Logged at the end of many code units, such as triggers, executeAnonymous, batch Apex message processing, @future methods, Apex test methods, Apex web service methods.
- Cumulative profiling information—Logged once at the end of the transaction. Contains information about the most expensive queries (that used the most resources), DML invocations, and so on.

The following is an example debug log:

```
23.0
APEX_CODE, DEBUG;APEX_PROFILING, INFO;CALLOUT, INFO;DB, INFO;SYSTEM, DEBUG;VALIDATION, INFO;VISUALFORCE, INFO;
WORKFLOW, INFO
11:47:46.030 (30064000) |EXECUTION_STARTED
11:47:46.030 (30159000) |CODE_UNIT_STARTED|[EXTERNAL]|TRIGGERS
11:47:46.030 (30271000) |CODE_UNIT_STARTED|[EXTERNAL]|01qD00000004JvP|myTrigger on Merchandise
trigger event BeforeUpdate for [001D000000IzMaE]
11:47:46.038 (38296000) |SYSTEM_METHOD_ENTRY|[2]|System.debug(ANY)
11:47:46.038 (38450000) |USER_DEBUG|[2]|DEBUG>Hello World!
11:47:46.038 (38520000) |SYSTEM_METHOD_EXIT|[2]|System.debug(ANY)
11:47:46.546 (38587000) |CUMULATIVE_LIMIT_USAGE
11:47:46.546|LIMIT_USAGE_FOR_NS|(default)|
    Number of SOQL queries: 0 out of 100
    Number of query rows: 0 out of 50000
    Number of SOSL queries: 0 out of 20
    Number of DML statements: 0 out of 150
    Number of DML rows: 0 out of 10000
    Number of script statements: 1 out of 200000
    Maximum heap size: 0 out of 6000000
    Number of callouts: 0 out of 10
    Number of Email Invocations: 0 out of 10
    Number of fields describes: 0 out of 100
    Number of record type describes: 0 out of 100
    Number of child relationships describes: 0 out of 100
    Number of picklist describes: 0 out of 100
    Number of future calls: 0 out of 10

11:47:46.546|CUMULATIVE_LIMIT_USAGE_END

11:47:46.038 (38715000) |CODE_UNIT_FINISHED|myTrigger on Merchandise trigger event BeforeUpdate
for [001D000000IzMaE]
11:47:47.154 (1154831000) |CODE_UNIT_FINISHED|TRIGGERS
11:47:47.154 (1154881000) |EXECUTION_FINISHED
```

Setting Debug Log Filters

Setting Debug Log Filters

User Permissions Needed	
To use the Developer Console:	“View All Data”
To use the execute anonymous text entry box:	“Author Apex”
To save changes to Apex classes and triggers:	“Author Apex”

When using the Developer Console or monitoring a debug log, you can specify the level of information that gets included in the log.

Log category

The [type of information logged](#), such as information from Apex or workflow rules.

Log level

The [amount of information logged](#).

Event type

The combination of log category and log level that specify which [events get logged](#). Each event can log additional information, such as the line and character number where the event started, fields associated with the event, duration of the event in milliseconds, and so on.

Debug Log Categories

You can specify the following log categories. The amount of information logged for each category depends on the [log level](#):

Log Category	Description
Database	Includes information about database activity, including every data manipulation language (DML) statement or inline SOQL or SOSL query.
Workflow	Includes information for workflow rules, such as the rule name, the actions taken, and so on.
Validation	Includes information about validation rules, such as the name of the rule, whether the rule evaluated true or false, and so on.
Callout	Includes the request-response XML that the server is sending and receiving from an external Web service. This is useful when debugging issues related to using Force.com Web services API calls.
Apex Code	Includes information about Apex code and can include information such as log messages generated by DML statements, inline SOQL or SOSL queries, the start and completion of any triggers, and the start and completion of any test method, and so on.
Apex Profiling	Includes cumulative profiling information, such as the limits for your namespace, the number of emails sent, and so on.
Visualforce	Includes information about Visualforce events including serialization and deserialization of the view state or the evaluation of a formula field in a Visualforce page.
System	Includes information about calls to all system methods such as the <code>System.debug</code> method.



Note: Visualforce isn't available in Database.com.

Debug Log Levels

You can specify the following log levels. The levels are listed from lowest to highest. [Specific events](#) are logged based on the combination of category and levels. Most events start being logged at the INFO level. The level is cumulative, that is, if you select FINE, the log will also include all events logged at DEBUG, INFO, WARN and ERROR levels.



Note: Not all levels are available for all categories: only the levels that correspond to one or more events.

- ERROR

- WARN
- INFO
- DEBUG
- FINE
- FINER
- FINEST

Debug Event Types

The following is an example of what is written to the debug log. The event is `USER_DEBUG`. The format is `timestamp | event identifier`:

- *timestamp*: consists of the time when the event occurred and a value between parentheses. The time is in the user's time zone and in the format `HH:mm:ss.SSS`. The value represents the time elapsed in nanoseconds since the start of the request. The elapsed time value is excluded from logs reviewed in the Developer Console.
- *event identifier*: consists of the specific event that triggered the debug log being written to, such as `SAVEPOINT_RESET` or `VALIDATION_RULE`, and any additional information logged with that event, such as the method name or the line and character number where the code was executed.

The following is an example of a debug log line.

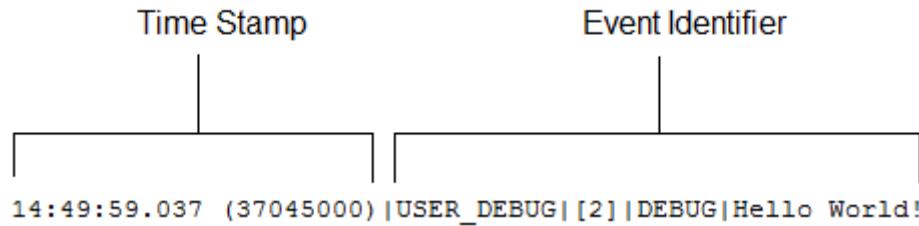


Figure 3: Debug Log Line Example

In this example, the event identifier is made up of the following:

- Event name:

```
USER_DEBUG
```

- Line number of the event in the code:

```
[2]
```

- Logging level the `System.Debug` method was set to:

```
DEBUG
```

- User-supplied string for the `System.Debug` method:

```
Hello world!
```

The following example of a log line is triggered by this code snippet.

```

1 @isTest
2 private class TestHandleProductPriceChange {
3     static testMethod void testPriceChange() {
4         Invoice_Statement__c invoice = new Invoice_Statement__c(status__c = 'Negotiating');
5         insert invoice;
6

```

Figure 4: Debug Log Line Code Snippet

The following log line is recorded when the test reaches line 5 in the code:

```
15:51:01.071 (55856000) |DML_BEGIN|[5]|Op:Insert|Type:Invoice_Statement__c|Rows:1
```

In this example, the event identifier is made up of the following:

- Event name:

```
DML_BEGIN
```

- Line number of the event in the code:

```
[5]
```

- DML operation type—Insert:

```
Op:Insert
```

- Object name:

```
Type:Invoice_Statement__c
```

- Number of rows passed into the DML operation:

```
Rows:1
```

The following table lists the event types that are logged, what fields or other information get logged with each event, as well as what combination of log level and category cause an event to be logged.

Event Name	Fields or Information Logged With Event	Category Logged	Level Logged
BULK_HEAP_ALLOCATE	Number of bytes allocated	Apex Code	FINEST
CALLOUT_REQUEST	Line number, request headers	Callout	INFO and above
CALLOUT_RESPONSE	Line number, response body	Callout	INFO and above
CODE_UNIT_FINISHED	None	Apex Code	ERROR and above
CODE_UNIT_STARTED	Line number, code unit name, such as MyTrigger on Invoice_Statement__c trigger event BeforeInsert for [new]	Apex Code	ERROR and above

Event Name	Fields or Information Logged With Event	Category Logged	Level Logged
CONSTRUCTOR_ENTRY	Line number, Apex class ID, the string <init>() with the types of parameters, if any, between the parentheses	Apex Code	DEBUG and above
CONSTRUCTOR_EXIT	Line number, the string <init>() with the types of parameters, if any, between the parentheses	Apex Code	DEBUG and above
CUMULATIVE_LIMIT_USAGE	None	Apex Profiling	INFO and above
CUMULATIVE_LIMIT_USAGE_END	None	Apex Profiling	INFO and above
CUMULATIVE_PROFILING	None	Apex Profiling	FINE and above
CUMULATIVE_PROFILING_BEGIN	None	Apex Profiling	FINE and above
CUMULATIVE_PROFILING_END	None	Apex Profiling	FINE and above
DML_BEGIN	Line number, operation (such as Insert, Update, and so on), record name or type, number of rows passed into DML operation	Apex Code	INFO and above
DML_END	Line number	Apex Code	INFO and above
EMAIL_QUEUE	Line number	Apex Code	INFO and above
EXCEPTION_THROWN	Line number, exception type, message	Apex Code	INFO and above
EXECUTION_FINISHED	None	Apex Code	ERROR and above
EXECUTION_STARTED	None	Apex Code	ERROR and above
FATAL_ERROR	Exception type, message, stack trace	Apex Code	ERROR and above
HEAP_ALLOCATE	Line number, number of bytes	Apex Code	FINEST and above
HEAP_DEALLOCATE	Line number, number of bytes deallocated	Apex Code	FINEST and above
IDEAS_QUERY_EXECUTE	Line number	DB	FINEST
LIMIT_USAGE_FOR_NS	Namespace, following limits: Number of SOQL queries Number of query rows Number of SOSL queries Number of DML statements Number of DML rows Number of script statements Maximum heap size Number of callouts Number of Email Invocations Number of fields describes	Apex Profiling	FINEST

Event Name	Fields or Information Logged With Event	Category Logged	Level Logged
	Number of record type describes Number of child relationships describes Number of picklist describes Number of future calls Number of find similar calls Number of System.runAs() invocations		
METHOD_ENTRY	Line number, the Force.com ID of the class, Apex Code method signature	DEBUG and above	
METHOD_EXIT	Line number, the Force.com ID of the class, Apex Code method signature. For constructors, the following information is logged: Line number, class name.	DEBUG and above	
POP_TRACE_FLAGS	Line number, the Force.com ID of the class or trigger that has its log filters set and that is going into scope, the name of this class or trigger, the log filter settings that are now in effect after leaving this scope	System	INFO and above
PUSH_TRACE_FLAGS	Line number, the Force.com ID of the class or trigger that has its log filters set and that is going out of scope, the name of this class or trigger, the log filter settings that are now in effect after entering this scope	System	INFO and above
QUERY_MORE_ITERATIONS	Line number, number of queryMore iterations	DB	INFO and above
SAVEPOINT_ROLLBACK	Line number, Savepoint name	DB	INFO and above
SAVEPOINT_SET	Line number, Savepoint name	DB	INFO and above
SLA_END	Number of cases, load time, processing time, Workflow number of case milestones to insert/update/delete, new trigger		INFO and above
SLA_EVAL_MILESTONE	Milestone ID	Workflow	INFO and above
SLA_NULL_START_DATE	None	Workflow	INFO and above
SLA_PROCESS_CASE	Case ID	Workflow	INFO and above
SOQL_EXECUTE_BEGIN	Line number, number of aggregations, query source	DB	INFO and above
SOQL_EXECUTE_END	Line number, number of rows, duration in milliseconds	DB	INFO and above

Event Name	Fields or Information Logged With Event	Category Logged	Level Logged
SOSL_EXECUTE_BEGIN	Line number, query source	DB	INFO and above
SOSL_EXECUTE_END	Line number, number of rows, duration in milliseconds	DB	INFO and above
STACK_FRAME_VARIABLE_LIST	Frame number, variable list of the form: <i>Variable number Value.</i> For example: var1:50 var2:'Hello World'	Apex Profiling	FINE and above
STATEMENT_EXECUTE	Line number	Apex Code	FINER and above
STATIC_VARIABLE_LIST	Variable list of the form: <i>Variable number Value.</i> For example: var1:50 var2:'Hello World'	Apex Profiling	FINE and above
SYSTEM_CONSTRUCTOR_ENTRY	Line number, the string <init>() with the System types of parameters, if any, between the parentheses		DEBUG
SYSTEM_CONSTRUCTOR_EXIT	Line number, the string <init>() with the System types of parameters, if any, between the parentheses		DEBUG
SYSTEM_METHOD_ENTRY	Line number, method signature	System	DEBUG
SYSTEM_METHOD_EXIT	Line number, method signature	System	DEBUG
SYSTEM_MODE_ENTER	Mode name	System	INFO and above
SYSTEM_MODE_EXIT	Mode name	System	INFO and above
TESTING_LIMITS	None	Apex Profiling	INFO and above
TOTAL_EMAIL_RECIPIENTS_QUEUED	Number of emails sent	Apex Profiling	FINE and above
USER_DEBUG	Line number, logging level, user-supplied string	Apex Code	DEBUG and above by default
VALIDATION_ERROR	Error message	Validation	INFO and above



Note: If the user sets the log level for the `System.Debug` method, the event is logged at that level instead.

Event Name	Fields or Information Logged With Event	Category Logged	Level Logged
VALIDATION_FAIL	None	Validation	INFO and above
VALIDATION_FORMULA	Formula source, values	Validation	INFO and above
VALIDATION_PASS	None	Validation	INFO and above
VALIDATION_RULE	Rule name	Validation	INFO and above
VARIABLE_ASSIGNMENT	Line number, variable name, a string representation of the variable's value, the variable's address	Apex Code	FINEST
VARIABLE_SCOPE_BEGIN	Line number, variable name, type, a value that indicates if the variable can be referenced, a value that indicates if the variable is static	Apex Code	FINEST
VARIABLE_SCOPE_END	None	Apex Code	FINEST
VF_APEX_CALL	Element name, method name, return type	Apex Code	INFO and above
VF_DESERIALIZE_VIEWSTATE_BEGIN	View state ID	Visualforce	INFO and above
VF_DESERIALIZE_VIEWSTATE_END	None	Visualforce	INFO and above
VF_EVALUATE_FORMULA_BEGIN	View state ID, formula	Visualforce	FINER and above
VF_EVALUATE_FORMULA_END	None	Visualforce	FINER and above
VF_PAGE_MESSAGE	Message text	Apex Code	INFO and above
VF_SERIALIZE_VIEWSTATE_BEGIN	View state ID	Visualforce	INFO and above
VF_SERIALIZE_VIEWSTATE_END	None	Visualforce	INFO and above
WF_ACTION	Action description	Workflow	INFO and above
WF_ACTION_TASK	Task subject, action ID, rule, owner, due date	Workflow	INFO and above
WF_ACTIONS_END	Summary of actions performed	Workflow	INFO and above
WF_APPROVAL	Transition type, EntityName: NameField Id, process node name	Workflow	INFO and above
WF_APPROVAL_REMOVE	EntityName: NameField Id	Workflow	INFO and above
WF_APPROVAL_SUBMIT	EntityName: NameField Id	Workflow	INFO and above
WF_ASSIGN	Owner, assignee template ID	Workflow	INFO and above
WF_CRITERIA_BEGIN	EntityName: NameField Id, rule name, Workflow rule ID, trigger type (if rule respects trigger types)	Workflow	INFO and above
WF_CRITERIA_END	Boolean value indicating success (true or false)	Workflow	INFO and above
WF_EMAIL_ALERT	Action ID, rule	Workflow	INFO and above
WF_EMAIL_SENT	Email template ID, recipients, CC emails	Workflow	INFO and above
WF_ENQUEUE_ACTIONS	Summary of actions enqueued	Workflow	INFO and above
WF_ESCALATION_ACTION	Case ID, business hours	Workflow	INFO and above

Event Name	Fields or Information Logged With Event	Category Logged	Level Logged
WF_ESCALATION_RULE	None	Workflow	INFO and above
WF_EVAL_ENTRY_CRITERIA	Process name, email template ID, Boolean value indicating result (true or false)	Workflow	INFO and above
WF_FIELD_UPDATE	EntityName: NameField Id, object or field name	Workflow	INFO and above
WF_FORMULA	Formula source, values	Workflow	INFO and above
WF_HARD_REJECT	None	Workflow	INFO and above
WF_NEXT_APPROVER	Owner, next owner type, field	Workflow	INFO and above
WF_NO_PROCESS_FOUND	None	Workflow	INFO and above
WF_OUTBOUND_MSG	EntityName: NameField Id, action ID, rule	Workflow	INFO and above
WF_PROCESS_NODE	Process name	Workflow	INFO and above
WF_REASSIGN_RECORD	EntityName: NameField Id, owner	Workflow	INFO and above
WF_RESPONSE_NOTIFY	Notifier name, notifier email, notifier template ID	Workflow	INFO and above
WF_RULE_ENTRY_ORDER	Integer, indicating order	Workflow	INFO and above
WF_RULE_EVAL_BEGIN	Rule type	Workflow	INFO and above
WF_RULE_EVAL_END	None	Workflow	INFO and above
WF_RULE_EVAL_VALUE	Value	Workflow	INFO and above
WF_RULE_FILTER	Filter criteria	Workflow	INFO and above
WF_RULE_INVOCATION	EntityName: NameField Id	Workflow	INFO and above
WF_RULE_NOT_EVALUATED	None	Workflow	INFO and above
WF_SOFT_REJECT	Process name	Workflow	INFO and above
WF_SPOOL_ACTION_BEGIN	Node type	Workflow	INFO and above
WF_TIME_TRIGGER	EntityName: NameField Id, time action, time action container, evaluation Datetime	Workflow	INFO and above
WF_TIME_TRIGGERS_BEGIN	None	Workflow	INFO and above

Debug Log Filtering for Apex Classes and Apex Triggers

Setting Debug Log Filters for Apex Classes and Triggers

Debug log filtering provides a mechanism for fine-tuning the log verbosity at the trigger and class level. This is especially helpful when debugging Apex logic. For example, to evaluate the output of a complex process, you can raise the log verbosity for a given class while turning off logging for other classes or triggers within a single request.

When you override the debug log levels for a class or trigger, these debug levels also apply to the class methods that your class or trigger calls and the triggers that get executed as a result. All class methods and triggers in the execution path inherit the debug log settings from their caller, unless they have these settings overridden.

The following diagram illustrates overriding debug log levels at the class and trigger level. For this scenario, suppose Class1 is causing some issues that you would like to take a closer look at. To this end, the debug log levels of Class1 are raised to the finest granularity. Class3 doesn't override these log levels, and therefore inherits the granular log filters of Class1. However, UtilityClass has already been tested and is known to work properly, so it has its log filters turned off. Similarly, Class2 isn't in the code path that causes a problem, therefore it has its logging minimized to log only errors for the Apex Code category. Trigger2 inherits these log settings from Class2.

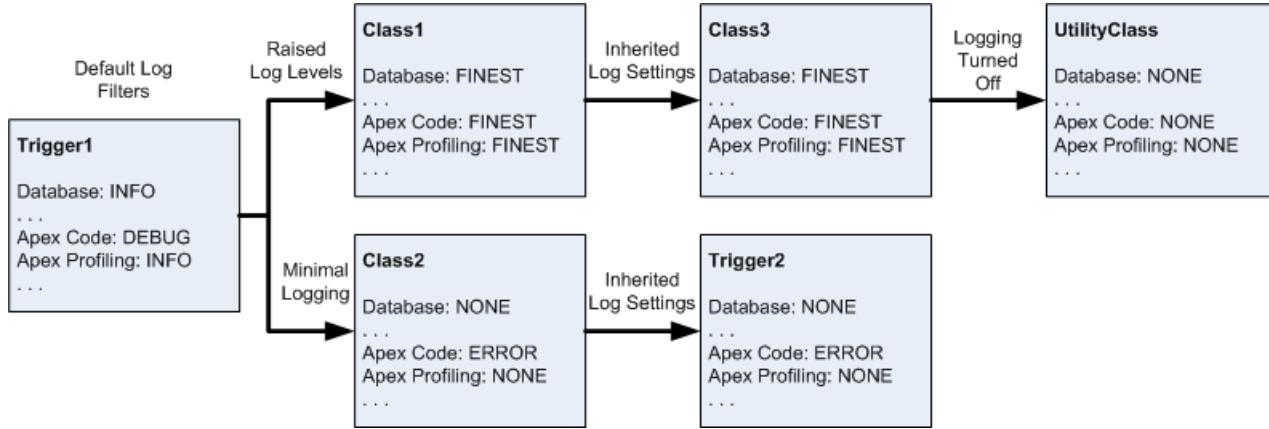


Figure 5: Fine-tuning debug logging for classes and triggers

The following is a pseudo-code example that the diagram is based on.

1. Trigger1 calls a method of Class1 and another method of Class2. For example:

```
trigger Trigger1 on Merchandise__c (before insert) {
    Class1.someMethod();
    Class2.anotherMethod();
}
```

2. Class1 calls a method of Class3, which in turn calls a method of a utility class. For example:

```
public class Class1 {
    public static void someMethod() {
        Class3.thirdMethod();
    }
}

public class Class3 {
    public static void thirdMethod() {
        UtilityClass.doSomething();
    }
}
```

3. Class2 causes a trigger, Trigger2, to be executed. For example:

```
public class Class2 {
    public static void anotherMethod() {
        // Some code that causes Trigger2 to be fired.
    }
}
```

To set log filters:

1. From a class or trigger detail page, click **Log Filters**.

2. Click **Override Log Filters.**

The log filters are set to the default log levels.

3. Choose the log level desired for each log category.

To learn more about debug log categories, debug log levels, and debug log events, see [Setting Debug Log Filters](#).

Collecting and Monitoring Debug Logs

See Also:

[Monitoring Debug Logs](#)

[Viewing Debug Logs](#)

Retaining Debug Logs

User Permissions Needed	
To view, retain, and delete debug logs:	“Manage Users”

You can retain and manage the debug logs for specific users.

The following are the limits for debug logs:

- Once a user is added, that user can record up to 20 debug logs. After a user reaches this limit, debug logs stop being recorded for that user. Click **Reset** on the Monitoring Debug logs page to reset the number of logs for that user back to 20. Any existing logs are not overwritten.
- Each debug log can only be 2 MB. Debug logs that are larger than 2 MB in size are truncated.
- Each organization can retain up to 50 MB of debug logs. Once your organization has reached 50 MB of debug logs, the oldest debug logs start being overwritten.

To specify that a user should have his or her debug logs retained:

- Click **Monitoring > Debug Logs**, then click **New**.
- Click the checkbox next to the user or users for whom you want to retain debug logs. Only users with a checkbox next to their names don't currently have their debug logs retained.
- Click **Save**.

Testing Apex

About Apex Unit Tests

User Permissions Needed
To define, edit, delete, set security, set version settings, show dependencies, and run tests “Author Apex” for Apex classes:

Testing is key to the success of your application, particularly if your application is to be deployed to customers. If you validate that your application works as expected and that there are no unexpected behaviors, your customers are going to trust you more.

To facilitate the development of robust, error-free code, Apex supports the creation and execution of *unit tests*. Unit tests are class methods that verify whether a particular piece of code is working properly. Unit test methods take no arguments, commit no data to the database, send no emails, and are flagged with the `testMethod` keyword in the method definition.

Before you can deploy your code, the following must be true:

- You must have at least 75% of your Apex covered by unit tests to deploy your code to production environments. In addition, all triggers must have some test coverage.
- We recommend that you have 100% of your code covered by unit tests, where possible.
- Calls to `System.debug` are not counted as part of Apex code coverage in unit tests.

You can run unit tests for:

- A specific class
- A subset of classes
- All unit tests in your organization

If your test calls another class or causes a trigger to execute, that Apex is included in the total amount used for calculating the percentage of code covered.

Working with Apex Test Execution

User Permissions Needed

To define, edit, delete, set security, set version settings, show dependencies, and run tests “Author Apex” for Apex classes:

To use the Apex Test Execution page:

1. Click **Develop > Apex Test Execution**.
2. Click **Select Tests....**
3. Select the tests to run. The list of tests contains classes that contain test methods.



Note: Classes whose tests are still running don't appear in the list.

4. Click **Run**.

After selecting test classes to run, the selected classes are placed in the Apex job queue for execution. The maximum number of test classes you can select for execution is the greater of 500 or 10 multiplied by the number of test classes in the organization.

While tests are running, you can select one or more tests and click **Abort** to cancel the test.

After a test finishes running, you can:

- Click the test to see result details. If a test fails, the first error message and the stack trace display.
- Click **View** to see the source Apex code.



Note: Test results display for 60 minutes after they finish running.

After you run tests using the Apex Test Execution page, you can display the percentage of code covered by those tests on the list of Apex classes. Click **Develop > Apex Classes**, then click **Calculate your organization's code coverage**.

You can also verify which lines of code are covered by tests for an individual class. Click **Develop > Apex Classes**, then click the percentage number in the Code Coverage column for a class.

Use the Apex Test Results page to see all test results for your organization. Click **Develop > Apex Test Execution > View Test History**.

Use the Developer Console to see additional information about your test execution:

1. Click **your Name > Developer Console**.
2. Run your tests using the Apex Test Execution page.
3. Check the Developer Console to step through the request.

Viewing Test Results

User Permissions Needed
To define, edit, delete, set security, set version settings, show dependencies, and run tests “Author Apex” for Apex classes:

To use the Apex Classes page to generate test results, click **Develop > Apex Classes**, then either click **Run All Tests** or click the name of a specific class that contains tests and click **Run Test**.

After you use the Apex Classes page to generate test results, the test result page contains the following sections. Each section can be expanded or collapsed.

- A summary section that details the number of tests run, the number of failures, the percentage of Apex code that is covered by unit tests, the total execution time in milliseconds, and a link to a downloadable debug log file.
- The debug log is automatically set to specific log levels and categories, which can't be changed.

Category	Level
Database	INFO
Apex Code	FINE
Apex Profiling	FINE
Workflow	FINEST
Validation	INFO



Important:

- ◊ You must have at least 75% of your Apex covered by unit tests to deploy your code to production environments. In addition, all triggers should have some test coverage.
 - ◊ We recommend that you have 100% of your code covered by unit tests, where possible.
 - ◊ Calls to `System.debug` are not counted as part of Apex code coverage in unit tests.
- Test successes, if any.
 - Test failures, if any.
 - A code coverage section.

This section lists all the classes and triggers in your organization, and the percentage of lines of code in each class and trigger that are covered by tests. If you click the coverage percent number, a page displays, highlighting all the lines of code for that class or trigger that are covered by tests in blue, as well as highlighting all the lines of code that are not covered by tests in red. It also lists how many times a particular line in the class or trigger was executed by the test.

- Test coverage warnings, if any.

You can also view test results by clicking **Develop > Apex Test Execution > View Test History**. This page displays all test results for your organization in the default view for 30 days unless cleared, not just tests that you have run.

To view test results of tests that are executed using the Apex Test Execution page, see [Working with Apex Test Execution](#) on page 404.

Apex Test Results

User Permissions Needed
To define, edit, delete, set security, set version settings, show dependencies, and run tests “Author Apex” for Apex classes:

Click **Develop > Apex Test Execution > View Test History** to view all test results for your organization, not just tests that you have run. Test results are retained for 30 days after they finish running, unless cleared.

To show a filtered list of items, select a predefined list from the View drop-down list, or click **Create New View** to define your own [custom view](#). To edit or delete any view you created, select it from the View drop-down list and click **Edit**.

Click **View** to view more details about a specific test run.

Apex Test Results Details

User Permissions Needed
To define, edit, delete, set security, set version settings, show dependencies, and run tests “Author Apex” for Apex classes:

To view all test results for your organization in the default view for 30 days unless cleared, not just tests that you have run, click **Develop > Apex Test Execution > View Test History**. Click **View** to view more details about a specific test run.

Deploying Apex

Deploying Apex Using Change Sets

User Permissions Needed
To use the Apex Deployment Tool: “Author Apex”

You can deploy Apex classes and triggers between connected organizations, for example, from a test database organization to your production organization. You can create an outbound change set in the Database.com user interface and add the Apex components that you would like to upload and deploy to the target organization. To learn more about change sets, see [Change Sets Overview](#).

Deploying Apex Using the Force.com Migration Tool

User Permissions Needed	
To use the Apex Deployment Tool:	“Author Apex”

Download the Force.com Migration Tool if you want to use a script for deploying Apex from a test database organization to a Database.com production organization using Apache's Ant build tool.

To download the Force.com Migration Tool:

1. Click **Develop > Tools**.
2. Click **Force.com Migration Tool**.
3. Save the `salesforce_ant.zip` file and unzip its contents to the location of your choice.

The `salesforce_ant.zip` file contains the files you need to run an ant task that exercises the `compileAndTest` API call, including:

- A `Readme.html` file that explains how to use the tools
- A Jar file containing the ant task: `ant-salesforce.jar`
- A sample folder containing:
 - ◊ A `codepkg\classes` folder that contains `SampleDeployClass.cls` and `SampleFailingTestClass.cls`
 - ◊ A `codepkg\triggers` folder that contains `SampleAccountTrigger.trigger`
 - ◊ A `mypkg\objects` folder that contains the custom objects used in the examples
 - ◊ A `removecodepkg` folder that contains XML files for removing the examples from your organization
 - ◊ A sample `build.properties` file that you must edit, specifying your credentials, in order to run the sample ant tasks in `build.xml`
 - ◊ A sample `build.xml` file, that exercises the `deploy` and `retrieve` API calls

For more information on the syntax and use of Apex, see the [Database.com Apex Code Developer's Guide](#).



Note: The Force.com Migration Tool is a free resource provided by salesforce.com to support its users and partners, but is not considered part of our Services for purposes of the salesforce.com Master Subscription Agreement.

Deploying Apex Using the Force.com IDE

User Permissions Needed	
To use the Apex Deployment Tool:	“Author Apex”

You can download the Force.com IDE to help you write Apex in projects that mirror your organization. Using this tool, you can also compile and test the code you write, synchronize changes between the organization and project, and deploy your code to another organization.



Note: The Force.com IDE is a free resource provided by salesforce.com to support its users and partners, but is not considered part of our Services for purposes of the salesforce.com Master Subscription Agreement.

Deploying Apex Using the Web Services API

User Permissions Needed	
To use the Apex Deployment Tool:	“Author Apex”

You can use the following Web services API to deploy your Apex to a test database organization.

- `compileAndTest()`
- `compileClasses()`
- `compileTriggers()`

All these calls take Apex code that contains the class or trigger, as well as the values for any fields that need to be set.

To learn more about these Web services API calls, see the [Web Services API and SOAP Headers for Apex appendix](#) in the [Database.com Apex Code Developer's Guide](#).

Scheduling and Monitoring Apex Jobs

See Also:

[Monitoring Scheduled Jobs](#)

[Monitoring the Apex Job Queue](#)

Scheduling Apex

Use the Apex scheduler if you have specific Apex classes that you want to run on a regular basis, or to run a batch Apex job using the Database.com user interface.

The scheduler runs as system: all classes are executed, whether the user has permission to execute the class or not. For more information about class permissions, see [Apex Class Security Overview](#) on page 384.



Important: Database.com only adds the process to the queue at the scheduled time. Actual execution may be delayed based on service availability.

To schedule jobs using the Apex scheduler:

1. Implement the [Schedulable interface](#) in an Apex class that instantiates the class you want to run.
2. Click **Develop > Apex Classes** and click **Schedule Apex**.
3. Specify the name of a class that you want to schedule.
4. Specify how often the Apex class is to run.
 - For **Weekly**—specify one or more days of the week the job is to run (such as Monday and Wednesday).
 - For **Monthly**—specify either the date the job is to run or the day (such as the second Saturday of every month.)
5. Specify the start and end dates for the Apex scheduled class. If you specify a single day, the job only runs once.
6. Specify a preferred start time. The exact time the job starts depends on what other jobs are in the queue at that time.
7. Click **Save**.



Note: You can only have ten active or scheduled jobs concurrently.

After you schedule an Apex job, you can monitor the progress of the job on the [All Scheduled Jobs](#) page.

Once the job has completed, you can see specifics about the job (such as whether it passed or failed, how long it took to process, the number of records process, and so on) on the [Apex Jobs](#) page.

Security Tips for Apex Development

Understanding Security

Apex enables adding custom functionality and business logic to databases in Database.com. However, as with any programming language, developers must be cognizant of potential security-related pitfalls.

Database.com incorporates several security defenses but careless developers can still bypass the built-in defenses in many cases and expose their applications and customers to security risks.

SOQL Injection

In other programming languages, the previous flaw is known as SQL injection. Apex does not use SQL, but uses its own database query language, SOQL. SOQL is much simpler and more limited in functionality than SQL. Therefore, the risks are much lower for SOQL injection than for SQL injection, but the attacks are nearly identical to traditional SQL injection. In summary SQL/SOQL injection involves taking user-supplied input and using those values in a dynamic SOQL query. If the input is not validated, it can include SOQL commands that effectively modify the SOQL statement and trick the application into performing unintended commands.

For more information on SQL Injection attacks see:

- http://www.owasp.org/index.php/SQL_injection
- http://www.owasp.org/index.php/Blind_SQL_Injection
- http://www.owasp.org/index.php/Guide_to_SQL_Injection
- <http://www.google.com/search?q=mysql+injection>

SOQL Injection Vulnerability in Apex

Below is a simple example of Apex code vulnerable to SOQL injection. The name variable contains a value supplied by the user. For example, this code could reside in a method of a custom Apex Web service where the name is a parameter of the method and is supplied by the client.

```
String qryString = 'SELECT Id FROM Merchandise__c WHERE ' +
    '(IsDeleted = false AND Name LIKE \'%' + name + '%\')';
sObject[] queryResult = Database.query(qryString);
```

This is a very simple example but illustrates the logic. The code is intended to search for merchandise items that have not been deleted. The user provides one input value called name. The value can be anything provided by the user and it is never validated. The SOQL query is built dynamically and then executed with the `Database.query` method. If the user provides a legitimate value, the statement executes as expected:

```
// User supplied value for name: Pencils
// Query string
SELECT Id FROM Merchandise__c WHERE (IsDeleted = false and Name like %Pencils%)
```

However, what if the user provides unexpected input, such as:

```
// User supplied value for name: test%' OR (Name LIKE '
```

In that case, the query string becomes:

```
SELECT Id FROM Merchandise__c WHERE (IsDeleted = false AND Name LIKE '%test%') OR (Name LIKE '%')
```

Now the results show all merchandise items, not just the non-deleted ones. A SOQL Injection flaw can be used to modify the intended logic of any vulnerable query.

SOQL Injection Defenses

To prevent a SOQL injection attack, avoid using dynamic SOQL queries. Instead, use static queries and binding variables. The vulnerable example above can be re-written using static SOQL as follows:

```
String queryName = '%' + name + '%';
sObject[] queryResult = [SELECT Id FROM Merchandise__c WHERE
    (IsDeleted = false AND Name LIKE :queryName)';
```

If you must use dynamic SOQL, use the `escapeSingleQuotes` method to sanitize user-supplied input. This method adds the escape character (\) to all single quotation marks in a string that is passed in from a user. The method ensures that all single quotation marks are treated as enclosing strings, instead of database commands.

Data Access Control

Database.com makes extensive use of data sharing rules. Each object has permissions and may have sharing settings for which users can read, create, edit, and delete.

When using an Apex class, the built-in user permissions and field-level security restrictions are not respected during execution. The default behavior is that an Apex class has the ability to read and update all data within the organization. Because these rules are not enforced, developers who use Apex must take care that they do not inadvertently expose sensitive data that would normally be hidden from users by user permissions, field-level security, or organization-wide defaults. For example, consider the following Apex pseudo-code:

```
public class MyClass {
    public void read() {
        Merchandise__c m = [SELECT Id FROM Merchandise__c WHERE Name = :value];
    }
}
```

In this case, all merchandise records are searched, even if the user currently logged in would not normally have permission to view these records. The solution is to use the qualifying keywords `with sharing` when declaring the class:

```
public with sharing class MyClass {
    . . .
}
```

The `with sharing` keyword directs the platform to use the security sharing permissions of the user currently logged in, rather than granting full access to all records.

Managing Chatter Functionality

Chatter API

Chatter is an application that helps people share business information securely and in real time. Users employ Chatter to share information, learn about their colleagues, connect with others, and keep up with the latest record and document updates.

You can add Chatter functionality to your apps using the REST-based Chatter API, which is optimized to work with Web 2.0 resources. The Chatter API makes it easy to add social functionality to applications that use Database.com. With the Chatter API, you can:

- Build a mobile client that displays a Chatter feed.
- Integrate a third-party Web application with Chatter so it can notify groups of users about events.
- Display a Chatter feed on an external system, such as an intranet site, after users are authenticated to your application.
- Make feeds actionable and integrated with third-party sites. For example, an app that posts a Chatter item to Twitter whenever the post includes #tweet hashtag.

Setting Up Chatter

User Permissions Needed	
To enable Chatter:	“Customize Application”

To set up Chatter, an administrator must:

1. [Enable and configure Chatter](#).
2. [Customize feed tracking](#) for objects and fields tracked in Chatter feeds.

Enabling, Disabling, and Configuring Chatter

User Permissions Needed	
To enable Chatter:	“Customize Application”

Enable Chatter to turn on Chatter feeds and groups so people using your application running on Database.com can collaborate with each other.

1. Click **Customize > Chatter > Settings**.
2. Click **Edit**.
3. Select **Enable**. Deselect to disable Chatter.
4. Click **Save**.

Consider the following tips when enabling Chatter:

- When Chatter is enabled for organizations with 15 or fewer users, all users automatically follow each other.

Customizing Chatter Feed Tracking

User Permissions Needed	
To customize fields tracked in feeds:	“Customize Application”
To view the field tracking setup page:	“View Setup and Configuration”

Feed tracking is available for objects and fields:

- Enable objects for feed tracking so people can follow records of that object type and see Chatter feed updates when records of that object type are created. For example, if you enable the Line Item object, people can follow line item records and see feed updates when they create line item records.
- Enable fields for feed tracking so people can see Chatter feed updates about changes to fields on records they follow. For example, if you enable the Unit Price field on line items, people see updates when Unit Price is changed on line items they follow.

[Sharing rules](#) and [field-level security](#) determine visibility of record changes in Chatter feeds; you must be able to see a record in order to see changes to that record in feeds. Tracked feed updates that are older than 90 days and have no likes or comments are deleted automatically.

You can configure feed tracking for users, Chatter groups, and custom objects.

- Click **Customize > Chatter > Feed Tracking**.
- Select an object.
- Select **Enable Feed Tracking**. The user and Chatter group objects don't have this checkbox.
- Select up to 20 fields to track.
- Click **Save**.
- Optionally, repeat steps 2 through 5 for additional objects.

Consider the following feed tracking tips:

- To stop tracking an object, deselect **Enable Feed Tracking**. You can't disable feed tracking for the user or Chatter group objects.
- You must track the owner field to enable a new record owner to [automatically follow that record](#).
- To restore the default feed tracking settings for an object, click **Restore Defaults**.
- The following standard field types can't be tracked:
 - ◊ Auto-number, formula, and roll-up summary fields
 - ◊ Encrypted and read-only system fields

Configuring Your Chatter Feed Settings

User Permissions Needed	
To view a record:	“Read” on the record

Automatically Following Records You Own

Because you automatically follow records you own, updates are sent to your Chatter feed when fields are changed on those records. To stop automatically following records you own:

- Click **My Chatter Settings > My Feeds**.
- Select Stop automatically following records.
- Click **Save**.

Managing Workflow

Understanding Workflow

Workflow Overview

User Permissions Needed	
To view workflow rules:	"View Setup and Configuration"
To create or change workflow rules:	"Customize Application"

Common Database.com operations such as record updates are part of an organization's standard processes. Instead of doing this work manually, you can [configure workflow](#) to do it automatically.

Begin by designing workflow rules and associating them with actions such as field updates or outbound messages.

About Workflow

Each workflow rule consists of:

- Criteria that determine when Database.com executes the workflow rule. Any change that causes a record to match this criteria can trigger the workflow rule—even changes to hidden fields.
- Immediate actions to take when the workflow rule executes.
- Time-dependent actions that Database.com queues when the workflow rule executes. When Database.com triggers a workflow rule that has time-dependent actions, you can use the [workflow queue](#) to monitor and cancel pending actions.

Workflow automates the following types of actions:

- Field Updates—Update the value of a field on a record.
- [Outbound Messages](#)—Send a secure configurable API message (in XML format) to a designated listener.

Workflow Terminology

Workflow Rule

A workflow rule sets workflow actions into motion when its designated conditions are met. You can configure workflow actions to execute immediately when a record meets the conditions in your workflow rule, or set time triggers that execute the workflow actions on a specific day. If a workflow action hasn't executed yet, you can view and modify it in the [Workflow Queue](#). See [Managing Workflow Rules](#) to get started using workflow rules.

Workflow Action

A workflow action is a field update or outbound message that fires when the conditions of a workflow rule are met.

Field Update

Field updates are workflow actions that specify the field you want updated and the new value for it. Depending on the type of field, you can choose to apply a specific value, make the value blank, or calculate a value based on a formula you create. To get started using field updates, see [Managing Field Updates](#).

Outbound Message

An outbound message is a workflow action that sends the information you specify to an endpoint you designate, such as an external service. An outbound message sends the data in the specified fields in the form of a SOAP message to the endpoint. To get started using outbound messages, see [Managing Outbound Messages](#).

Managing Workflow Rules

Creating a Workflow Rule

Creating Workflow Rules

User Permissions Needed

To create or change workflow rules and actions: “Customize Application”



Watch a Demo (2:50 minutes)

Automate your organization's standard processes by configuring workflow rules.

To create a new rule:

1. Select the object to which the workflow rule applies.
2. Configure the workflow rule settings and criteria.
3. Configure the workflow actions.
4. Activate the workflow rule.

Selecting the Object for Your Workflow Rule

User Permissions Needed

To create or change workflow rules and actions: “Customize Application”

1. Click **Create > Workflow & Approvals > Workflow Rules**.
2. On the workflow rules list page, click **New Rule**.
3. Choose an object to which you want this workflow rule to apply.



Note:

- If you have a workflow action that updates a field on a related object, that target object isn't the one associated with the workflow rule.

4. Click **Next**.

Configuring Rule Settings and Criteria

User Permissions Needed

To create or change workflow rules and actions: “Customize Application”

After selecting the object for your workflow rule, you can configure its settings and criteria.

1. Enter a rule name.
2. Enter a description for the rule.
3. Choose the evaluation criteria:
 - When a record is created, or when a record is edited and did not previously meet the rule criteria: Choose this option to include new records and updates to existing records, unless the rule just ran and still meets the rule criteria. The rule isn't re-triggered on edits that don't affect rule criteria.
 - Only when a record is created: Choose this option to ignore updates to existing records.

- Every time a record is created or edited: Choose this option to include new records and updates to existing records and repeatedly trigger the rule, even if the record still meets the criteria.



Note: You can't add time-dependent actions to a rule if you choose Every time a record is created or edited.

4. Enter your rule criteria:

- Choose criteria are met and select the filter criteria that a record must meet to trigger the rule. If your organization uses multiple languages, enter filter values in your organization's default language. You can add up to 25 filter criteria, of up to 255 characters each.



Note: Workflow rules respect the user's locale and aren't triggered when the user is in a different language than that of the organization.

- Choose formula evaluates to true and enter a formula that returns a value of "True" or "False." Database.com triggers the rule if the formula returns "True." Examples of useful workflow formulas include:
 - ◊ If the number of filled positions equals the number of total positions on a job, update the Job Status field to "Filled."
 - ◊ If mileage expenses associated with visiting a customer site are 35 cents per mile and exceed a \$1,000 limit, automatically update the Approval Required field to "Required."

Some functions aren't available in workflow rule formulas.



Tip: You can use merge fields for directly related objects in workflow rule formulas. For more information, see [Merge Fields Overview](#).

5. Click **Save & Next**.

Configuring Workflow Actions

User Permissions Needed	
To create or change workflow rules and actions:	"Customize Application"

Add immediate and time-dependent actions to the workflow rule. Immediate actions trigger once evaluation criteria are met. Time-dependent actions specify when Database.com executes the workflow action.



Tip: Time-dependent actions and time triggers are complex features with several [considerations](#).

If you plan on configuring workflow rules that have [time-dependent actions](#), specify a default workflow user.

Database.com associates the default workflow user with a workflow rule if the user who initiated the rule is no longer active.

1. To add an immediate workflow action, click **Add Workflow Action** in the Immediate Workflow Actions section and select:

- New Field Update to define a [field update](#) to associate with the rule
- New Outbound Message to define an [outbound message](#) to associate with the rule

- Select Existing Action to select an [existing action](#) to associate with the rule
2. To add a time-dependent workflow action, click **Add Time Trigger** in the Time-Dependent Workflow Actions section and:
- a. Specify a number of days or hours before or after a date relevant to the record, such as the date the record was created or modified. If the workflow rule is in effect when this time occurs, the time trigger fires the workflow action.
 - b. Click **Save**.
-  **Note:** The **Add Time Trigger** button is unavailable if:
- The rule criteria is set to Every time a record is created or edited
 - The rule is activated
 - The rule is deactivated but has [pending actions](#) in the workflow queue
3. Configure additional immediate or time-dependent actions.
4. Click **Done**.

 **Note:** For all custom objects, you can create workflow actions where a change to a detail record updates a field on the related master record. Cross-object field updates work for custom-to-custom master-detail relationships. For more information, see [Understanding Cross-Object Field Updates](#) on page 417.

Activating a Workflow Rule

User Permissions Needed	
To create or change workflow rules and actions:	“Customize Application”

Database.com doesn't trigger a workflow rule until you activate it. To activate a workflow rule, click **Activate** on the workflow rule detail page. Click **Deactivate** to prevent a rule from triggering or if you want to edit the time-dependent actions and time triggers associated with the rule.

Managing Workflow Actions

[Creating a Workflow Action](#)
[Managing Field Updates](#)
[Defining Field Updates](#)

User Permissions Needed	
To define, edit, or delete field updates:	“Customize Application”

Field updates allow you to automatically specify a field value. Field updates are actions associated with workflow rules. Before you begin, review [Field Update Considerations](#) on page 418.

 **Note:** For all custom objects, you can create workflow actions where a change to a detail record updates a field on the related master record. Cross-object field updates work for custom-to-custom master-detail relationships. For more information, see [Understanding Cross-Object Field Updates](#) on page 417.

1. Click **Create > Workflow & Approvals > Field Updates**.
2. Click **New Field Update**.
3. Configure the field update.

- a. Enter a name for this field update.
- b. Enter a unique name to refer to this component in the API. The requirement for uniqueness is only within the selected object type. You can have actions of the same type with the same unique name, provided they are defined for different objects. The Unique Name field can contain only underscores and alphanumeric characters. It must be unique within the selected object type, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
- c. Enter a description.
- d. Define the field update, including the object associated with the workflow rule and the field to update. Note that the field to update may be on a related object in a master-detail relationship. Fields are shown only for the target object that you select.



Note: You can use field updates on encrypted custom fields, but if you try to use a formula to set the new value, the encrypted field isn't available in the formula editor.

- e. Select the Re-evaluate Workflow Rules After Field Change checkbox if you want workflow rules on this object to be re-evaluated after the field value is updated. For more information, see [Field Updates That Re-evaluate Workflow Rules](#) on page 418.
- f. Specify the new field value. The available options depend on the type of field you are updating. You can:
 - Choose **A specific value**, and enter the value in the space provided.
 - Choose **A blank value (null)** if you want Database.com to remove any existing value and leave the field blank. This option isn't available for required fields, checkboxes, and some other types of fields.
 - For record owners, choose the user to whom the record should be assigned. For custom object records, you can also choose a queue for this field. Select **Notify Assignee** to send an email to the new record owner.
 - For checkboxes, choose **True** to select the checkbox and **False** to deselect it.
 - For picklists, select a specific value from the drop-down list, or select the value above or below the current value based on the sorting specified in the picklist definition. If you sort values alphabetically, the values above or below may be different for users in other languages.
 - Choose **Use a formula to set the new value** to calculate the value based on an expression, merge fields, or other values.
 - ◊ If you're building a formula for workflow or validation rules, click **Insert Field**, choose a field, and click **Insert**.
 - ◊ To insert an operator, choose the appropriate operator icon from the **Insert Operator** drop-down list. Use the examples in [Operators and Functions](#) on page 132.
 - ◊ You can insert a function from the Functions list. Functions are pre-built formulas that you can customize with your input parameters. See [Operators and Functions](#) for a description of each operator and function.
- g. Click **Save**.

Understanding Cross-Object Field Updates

For all custom objects, you can create workflow actions where a change to a detail record updates a field on the related master record. Cross-object field updates work for custom-to-custom master-detail relationships. For example, in a custom recruiting application, create a workflow rule that sets the status of an application (the master object) to “Closed” when a candidate (the detail object) accepts the job.



Note: The cross-object field update function may depend on certain critical updates being active. Click **Critical Updates** to see if your organization requires any updates.

Custom Object to Custom Object

You can use cross-object field updates for all custom objects that are children of custom objects in a master-detail relationship.

Field Updates That Re-evaluate Workflow Rules

Selecting the **Re-evaluate Workflow Rules After Field Change** checkbox on the Field Update Edit page allows you to set a workflow field update to re-evaluate all workflow rules on the object if the field update results in a change to the value of the field.

- 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.

Field Update Considerations

User Permissions Needed	
To define, edit, or delete field updates:	“Customize Application”

When creating field updates for **workflow rules**, consider the following:

Field Update Processing

- Field updates occur before outbound messages.
- For all custom objects, you can create workflow actions where a change to a detail record updates a field on the related master record. Cross-object field updates work for custom-to-custom master-detail relationships. For more information, see [Understanding Cross-Object Field Updates](#) on page 417.
- The cross-object field update function may depend on certain critical updates being active. Click **Critical Updates** to see if your organization requires any updates.
- Field updates function independently of field-level security. Therefore, a workflow rule can update fields even though they are hidden on the user's page layout.
- The result of a field update is unpredictable when a single workflow rule includes multiple field updates that apply different values to the same field.
- Field updates may affect the information in a related list.
- If a user gets a field update error when saving a record, you can use the [debug log](#) to see which field update failed. The debug log stops when a failure occurs.
- If your organization uses multiple currencies, currency fields are updated using the record's currency. If you choose to update a field based on a formula, any values in your formula are interpreted in the currency of the record.
- Before changing a custom field's type, make sure that it isn't the target of a field update or referenced in a field update formula that would be invalidated by the new type.
- Field updates are tracked in the History related list if you have [set history tracking](#) on those fields.

- Because updates to records based on workflow rules don't trigger validation rules, workflow rules can invalidate previously valid fields.

Field Update Limitations

- The results of a field update can't trigger additional rules such as validation, assignment, auto-response, or escalation rules.
- The results of a field update can trigger additional workflow rules if you have flagged the field update to do so. For more information, see [Field Updates That Re-evaluate Workflow Rules](#) on page 418.
- Read-only fields like formula or auto-number fields aren't available for field updates.
- The Language picklist field on multilingual solutions isn't available for field updates.
- Field update actions that update fields on related objects are supported only for use with workflow rules.
- If a field update references a specific user, you can't deactivate that user. For example, if your field update is designed to change the owner of a record to Bob Smith, change the field update before deactivating Bob Smith.
- You can't delete a custom field that is referenced by a field update.
- Some fields aren't available for field updates.
- You can update long text area fields, but the option to insert A specific value restricts you to entering up to 255 characters.
- You can't make a field universally required if it's used by a field update that sets the field to a blank value. For details, see [About Universally Required Fields](#) on page 193.
- You can use field updates on encrypted custom fields, but if you try to use a formula to set the new value, the encrypted field isn't available in the formula editor.

 **Tip:** Workflow field updates that run based on a time-dependent action don't trigger any rules.

[Managing Outbound Messages](#)
[Defining Outbound Messages](#)

User Permissions Needed	
To define, edit, or delete outbound messages:	<p>“Customize Application”</p> <p>AND</p> <p>“Manage Translation”</p> <p>AND</p> <p>“Manage Territories” (if territories are enabled)</p>

An outbound message is a workflow action that sends the information you specify to an endpoint you designate, such as an external service. An outbound message sends the data in the specified fields in the form of a SOAP message to the endpoint.

When you associate an outbound message with a workflow rule, the outbound message sends the selected information to the associated endpoint URL when the rule is triggered. Once the endpoint URL receives the message, it can take the information from the message and process it, assuming that the Web service has been configured appropriately.

For security reasons, Database.com restricts the outbound ports you may specify to one of the following:

- 80: This port only accepts HTTP connections.
- 443: This port only accepts HTTPS connections.
- 1024–66535 (inclusive): These ports accept HTTP or HTTPS connections.



Note: Outbound messages can't be associated with workflow rules on custom junction objects.

To define outbound messages:

1. Click **Create > Workflow & Approvals > Outbound Messages**.
2. Click **New Outbound Message**.



Note: If you don't have this option, your organization doesn't have outbound messages enabled. Contact salesforce.com to enable outbound messages.

3. Choose the object that has the information you want included in the outbound message, and click **Next**.
4. Configure the outbound message.
 - a. Enter a name for this outbound message.
 - b. Enter a unique name to refer to this component in the API. The requirement for uniqueness is only within the selected object type. You can have actions of the same type with the same unique name, provided they are defined for different objects. The **Unique Name** field can contain only underscores and alphanumeric characters. It must be unique within the selected object type, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
 - c. Enter a description.
 - d. Enter an endpoint URL for the recipient of the message. Database.com sends a SOAP message to this endpoint.
 - e. Select the Database.com user to use when sending the message. The chosen user controls data visibility for the message that is sent to the endpoint.
 - f. Select **Send Session ID** if you want the Database.com session ID included in the outbound message. Include the session ID in your message if you intend to make API calls and you don't want to include a username and password. Never send a username and password in an unencrypted message, especially in a production environment. It isn't secure.
 - g. Select **Add failures to failed outbound message related list** if you want this message to move to the failed outbound messages related list after all retries have been attempted. If you don't see this option, failed outbound message notifications aren't enabled for your organization. Contact your salesforce.com representative.
 - h. Select the fields to include in the outbound message and click **Add**.
5. Click **Save**, and review the outbound message detail page:
 - The **API Version** field is automatically generated and set to the current API version when the outbound message was created. This API version is used in API calls back to Database.com using the enterprise or partner WSDLs. The **API Version** can only be modified by using the Metadata API.
 - Click the **Click for WSDL** link to view the WSDL associated with this message.

The WSDL is bound to the outbound message and contains the instructions about how to reach the endpoint service and what data is sent to it.
6. If your endpoint uses a client certificate, right-click the **Download Client Certificate** link on the outbound message detail page, and save the certificate to the appropriate location. Then you can import the downloaded certificate into your application server and configure your application server to request the client certificate.
7. To put outbound messages into action, associate them with a [workflow rule](#).

Outbound Message Notifications

User Permissions Needed

To view or edit outbound message notification settings: “Customize Application”

You can request that up to five users receive a notification listing all outbound messages that have failed for at least 24 hours. A fresh notification is sent every 24 hours until you cancel the request.

Failed messages are deleted from the failed outbound messages related list after seven days. Before they are removed, you can delete them yourself or request that they be retried again.

To view the current outbound message notification requests, click **Monitoring > Outbound Message Notifications**.

You can perform several tasks here:

- Edit a notification.
- Delete a notification. Since you can only create five, you may need to delete one before you can create more.
- Create a new notification request.



Note: If you don't have this option, your organization doesn't have outbound messages enabled. Contact salesforce.com to enable outbound messages.

Creating and Editing Outbound Message Notifications

User Permissions Needed

To view or edit outbound message notification settings: “Modify All Data”

You can request that up to five users receive a notification listing all outbound messages that have failed for at least 24 hours. A fresh notification is sent every 24 hours until you cancel the request.

To create a notification request:

1. Click **Monitoring > Outbound Message Notifications**.
2. Click **New**.
3. Enter a full username, or click the icon to select it from a list of usernames.
4. Click **Save** to save the request, or **Save & New** to save this request and create a new one.



Note: If you don't have this option, your organization doesn't have outbound messages enabled. Contact salesforce.com to enable outbound messages.

Tracking Outbound Message Delivery Status

User Permissions Needed

To track outbound messages: “Modify All Data”

An outbound message is a workflow action that sends the information you specify to an endpoint you designate, such as an external service. An outbound message sends the data in the specified fields in the form of a SOAP message to the endpoint.

To track the status of an outbound message, click **Monitoring > Outbound Messages**. Alternatively, click **Create > Workflow & Approvals > Outbound Messages**, and then click **View Message Delivery Status**. Here you can view the status of your outbound messages:

- The **Next items for delivery** related list contains all outbound messages that are awaiting delivery.
- The **Oldest failures** in queue related list contains the oldest failures that have not yet been deleted (because they have not been delivered and are not 24 hours old).
- The **Failed** outbound messages related list contains all the outbound messages that failed to be delivered and are no longer being retried. Messages are listed here only if you configure the message when you create it by selecting **Add failures to failed outbound message related list**. For more information, see [Defining Outbound Messages](#) on page 419. If you do not see this related list, it has not been enabled for your organization. Contact your salesforce.com representative.

You can perform several tasks here:

- Click **Refresh** to refresh the list.
- Click any workflow action ID to view the action that triggered it.
- Click **Retry** to change the **Next Attempt** date to now. This causes the message delivery to be immediately retried. If you select **Retry** in the **Failed outbound messages** related list, the outbound message moves to the **Next items for delivery** related list and is retried for another 24 hours.
- Click **Del** to permanently remove the outbound message from the queue.



Note: If you don't have this option, your organization doesn't have outbound messages enabled. Contact salesforce.com to enable outbound messages.

Viewing Outbound Message Notification Requests

User Permissions Needed	
To view or edit outbound message notification settings:	“Modify All Data”

You can request that up to five users receive a notification listing all outbound messages that have failed for at least 24 hours. A fresh notification is sent every 24 hours until you cancel the request.

This page is displayed when you have saved a new notification request. You can:

- Click **Edit** to change the username for a notification request. This is simpler than deleting the request and then creating a new one.
- Click **Delete** to delete the notification request.
- Click **Clone** to create a new notification request with the same username.

Selecting Existing Actions

User Permissions Needed	
To select existing actions:	“Customize Application”

Workflow actions created for one [workflow rule](#) can be associated with other workflow rules.

To associate existing workflow actions with a workflow rule:

1. Click **Create > Workflow & Approvals > Workflow Rules**.
2. Select the workflow rule.
3. Click **Edit** in the Workflow Actions section.
4. Click **Add Workflow Action** in either the Immediate Workflow Actions or Time-Dependent Actions section, depending on when you want the action to occur, and choose **Select Existing Action**. See [Configuring Workflow Actions](#) on page 415.
5. Select the type of action to associate with the workflow rule. The **Available Actions** box lists all existing actions of that type.
6. Enter the name of a specific action in the text field and click **Find**.
7. Select the actions in the **Available Actions** box and use the right arrow to move them to the **Selected Actions** box. If necessary, select the left arrow to remove actions from the **Available Actions** box.
8. Click **Save**.

Time-Dependent Action and Time Trigger Considerations

User Permissions Needed
To define, edit, or delete time-dependent actions and time triggers: “Customize Application”

When creating time-dependent actions and time triggers for [workflow rules](#), consider the following:

Defining Time Triggers

- When defining a time trigger, use standard and custom date and date/time fields defined for the object. Specify time using days and hours.
- You can add actions to existing time triggers.

Time Trigger Processing

- Time-dependent actions aren't executed independently. They're processed several times every hour, where they're grouped together and executed as a single batch. Therefore, any triggers that fire as a result of those grouped actions are also executed in a single batch. This behavior can cause you to exceed your Apex governor limits if you design your time-based workflow in conjunction with Apex triggers.
- Database.com evaluates time-based workflow on the organization's time zone, not the users'. Users in different time zones may see differences in behavior.
- Database.com doesn't necessarily execute time triggers in the order they appear on the workflow rule detail page. Workflow rules list time triggers that use the **Before** field first, followed by time triggers that use the **After** field.
- Database.com doesn't display time-dependent action controls on the workflow rule edit page if you set the workflow rule evaluation criteria to **Every time a record is created or updated**.
- If you change a date field that is referenced by an unfired time trigger in a workflow rule that has been evaluated, Database.com recalculates the unfired time triggers associated with the rule. If Database.com recalculates the time triggers to a date in the past, Database.com triggers the associated actions shortly after you save the record.
- If a workflow rule has a time trigger set for a time in the past, Database.com queues the associated time-dependent actions to execute sometime within the hour, depending on system usage.
- Time-dependent actions remain in the [workflow queue](#) only as long as the workflow rule criteria are still valid. If a record no longer matches the rule criteria, Database.com removes the time-dependent actions queued for that record.
- Database.com ignores time triggers that reference null fields.

- Time-dependent actions can automatically be queued again if the record is updated and the evaluation criteria is When a record is created, or when a record is edited and did not previously meet the entry criteria.
- Deleting a record that has pending actions removes the pending actions from the [workflow queue](#). You can't restore the actions, even if you undelete the record.
- If the evaluation criteria is Only when a record is created, the workflow rule evaluates its time triggers only once. If the record that fired the rule changes to no longer meet the evaluation criteria, Database.com removes the pending actions from the queue and never reapplies the rule to the record.
- You can deactivate a workflow rule at any time. If the rule has pending actions in the [workflow queue](#), editing the record that triggered the rule removes the pending actions from the queue. If you don't edit the record, the pending actions are processed even though the rule has been deactivated.
- Time-based workflow rules aren't reevaluated if an Apex post trigger which is executed as a result of a workflow rule updates a field so its value no longer meets the time-based workflow rule criteria.
- Configuring the Due Date to "Rule Trigger Date" sets time triggers and workflow task due dates relative to when the workflow rule evaluation criteria are met.
- You can add a new active workflow rule with time triggers in a change set and deploy it. You can only change time triggers on a workflow rule in a change set if it's inactive. The rule must be activated in the destination organization manually or through another change set that only activates workflow rules and makes no time trigger changes.

For example, if you have an inactive workflow rule in your destination organization and your change set contains an active workflow rule with the same name and new or different time triggers, the deployment will fail because it will activate the workflow rule first, then try to add or remove time triggers.



Note: Time-dependent actions must be added manually when including a workflow rule in a change set. The **View/Add Dependencies** function doesn't detect time-dependent actions.

Limitations

- Time triggers don't support minutes or seconds.
- Time triggers can't reference the following:
 - ◊ DATE or DATETIME fields containing automatically derived functions, such as TODAY or NOW
 - ◊ Formula fields that include related-object merge fields
- Database.com limits the number of time triggers an organization can execute per hour. If an organization exceeds the limits for its Edition, Database.com defers the execution of the additional time triggers to the next hour.
- You can't add or remove time triggers if:
 - ◊ The workflow rule is active.
 - ◊ The workflow rule is deactivated but has pending actions in the queue.
 - ◊ The workflow rule trigger type is set to Every time a record is created or edited.

Notes on Using Workflow Rules

User Permissions Needed	
To view workflow rules:	"View Setup and Configuration"
To create or change workflow rules:	"Customize Application"

Consider the following when configuring workflow rules:

- Each workflow rule applies to a single object. For the maximum number of workflow rules allowed on an object, see [Database.com Limits](#).
- For all custom objects, you can create workflow actions where a change to a detail record updates a field on the related master record. Cross-object field updates work for custom-to-custom master-detail relationships. For more information, see [Understanding Cross-Object Field Updates](#) on page 417.
- The cross-object field update function may depend on certain critical updates being active. Click **Critical Updates** to see if your organization requires any updates.
- Workflow rules can be triggered any time a record is saved or created, depending on your rule criteria. However, rules created after saving records aren't triggered by those records retroactively.
- Workflow rules on custom objects are automatically deleted if the custom object is deleted.
- Workflow rules trigger automatically and are invisible to the user.
- The order in which actions are executed is not guaranteed. Field update actions are executed first, followed by other actions.
- Saving or creating records can trigger more than one rule.
- The following actions don't trigger workflow rules:
 - ◊ Mass replacing picklist values
 - ◊ Mass updating address fields
- Workflow rules respect the user's locale and aren't triggered when the user is in a different language than that of the organization.
- If you use record types in your workflow rule criteria whose labels have been translated using the translation workbench, the translated label value won't trigger the workflow rule. Workflow criteria evaluate the master label value and ignore the translated value. To avoid this problem, set the workflow criteria to evaluate the master record type label value by entering it manually in the Value field.

Workflow Rule Limitations

- You can't create outbound messages for workflow rules on [junction objects](#).



Tip: You can use the Developer Console to debug workflow rules. The Developer Console lets you [view debug log details](#) and [information about workflow rules and actions](#), such as the name of the user who triggered the workflow rule and the name and ID of the record being evaluated.

Developing with APIs

Which API Should I Use?

User Permissions Needed	
To use the API	“API Enabled”

Salesforce.com provides programmatic access to your organization's information using simple, powerful, and secure application programming interfaces.

When to Use SOAP API

You can use SOAP API to create, retrieve, update or delete records. With more than 20 different calls, SOAP API also allows you to maintain passwords, perform searches, and much more. Use SOAP API in any language that supports Web services.

When to Use REST API

REST API provides a powerful, convenient, and simple Web services interface for interacting with Database.com. Its advantages include ease of integration and development, and it is an excellent choice of technology for use with mobile applications and Web 2.0 projects. However, if you have large numbers of records to process, you may wish to use [Bulk API](#), which is based on [REST](#) principles and optimized for large sets of data.

When to Use Bulk API

Bulk API is based on REST principles, and is optimized for loading or deleting large sets of data. It allows you to query, insert, update, upsert, or delete a large number of records asynchronously by submitting a number of batches which are processed in the background by Database.com.

SOAP API, in contrast, is optimized for real-time client applications that update small numbers of records at a time. Although SOAP API can also be used for processing large numbers of records, when the data sets contain hundreds of thousands of records it becomes less practical. Bulk API is designed to make it simple to process data from a few thousand to millions of records.

The easiest way to use Bulk API is to enable it for processing records in Data Loader using CSV files. This avoids the need to write your own client application.

When to Use Metadata API

Use Metadata API to retrieve, deploy, create, update or delete customization information for your organization. The most common use is to migrate changes from a test database or testing organization to your production organization. Metadata API is intended for managing customizations and for building tools that can manage the metadata model, not the data itself. To create, retrieve, update or delete records use Metadata API to manage your data.

The easiest way to access the functionality in Metadata API is to use the Force.com IDE or Force.com Migration Tool. These tools are built on top of Metadata API and use the standard Eclipse and Ant tools respectively to simplify the task of working with Metadata API. Built on the Eclipse platform, the Force.com IDE provides a comfortable environment for programmers familiar with integrated development environments, allowing you to code, compile, test, and deploy all from within the IDE itself. The Force.com Migration Tool is ideal if you want to use a script or a command-line utility for moving metadata between a local directory and a Database.com organization.

When to Use Chatter API

The Chatter API is a REST API that provides programmatic access to Chatter feeds and social data such as users, groups, followers, and files. It's used by developers who want to integrate Chatter into a variety of applications such as mobile applications, intranet sites, and third-party Web applications. The Chatter API is similar to APIs offered by other companies with feeds, such as Facebook and Twitter. Its advantages include ease of integration and development.

Monitoring API Usage

About API Usage Notifications

User Permissions Needed	
To view, create, edit, or delete notifications:	“API Enabled”

When you create a request usage notification, you specify an administrator to receive an email notification whenever your organization exceeds a specified limit for the number of API requests made in a specified span of hours. For more information on API request limits, see “Force.com API Usage Limits” in the online help.

To view existing API usage notifications, click **Monitoring > API Usage Notifications**.

From the notifications list, you can do any of the following:

- Click **Edit** or **Del** to edit or delete an existing notification.
- View the name of the user who will receive the notification.
- View the notification interval, which defines the frequency at which the notifications are sent. For example, if the notification interval is four hours, a notification will be sent only if the last notification was sent at least four hours ago. Thus, during a 24-hour period, a maximum of six notifications will be sent.
- View the percent of the limit which, if exceeded, triggers a notification to be sent. For example, if your organization has a limit of 1,000,000 requests, and you set a threshold percentage of 60 (60%) and a notification interval of 24 hours, when 600,000 API requests have been sent in a 24-hour period, the specified user receives a notification.
- View the name of the user who created the notification and when the notification was created, as well as the last time the notification was modified, and the name of the user who made the modification.

To create a new notification, click **New**.

You can create up to ten notifications per organization.

Creating and Editing API Usage Notifications

User Permissions Needed	
To view, create, edit, or delete notifications:	“API Enabled”

On the API usage metering edit page accessible via **Monitoring > API Usage Notifications**, you can supply the required values for a rate-limiting notification:

- The Database.com user who will receive the notifications.
- The threshold percentage—the percentage of the rate limit that, once exceeded in the specified notification interval, triggers a notification to be sent to the specified user. Value must be between 0 and 100.
- The time period for which the number of requests is measured, in hours. For example, if the interval is 24, the rate must be exceeded in the past 24 hours for a notification to be sent.

If you change the time period, the new time period does not take effect until after the next notification of the existing time period. For example, assume you have set the time period to send notifications every hour. Then at 4:05 p.m., you set the time period to send notifications every 24 hours. A last notification from the old time period is sent at 5:00 p.m.. The next notification would be sent at 5:00 p.m. the next day.

Viewing API Usage Notifications

User Permissions Needed	
To view, create, edit, or delete notifications:	“API Enabled”

On the API usage notifications detail page, you can view information about a notification:

- Notification Recipient—The username for the person to whom the email notification is sent.
- Threshold—The percent of the usage limit that, when reached, triggers an email notification.
- Notification Interval (Hours)—The frequency at which the notifications are sent. For example, if the notification interval is four hours, a notification is sent only if the last notification was sent at least four hours ago. Thus, during a 24-hour period, a maximum of six notifications will be sent.
- Created By—The user who created the notification request, and the time it was created.
- Modified By—The user who last edited the notification.

On this page, you can also create a new notification based on the values of the notification being displayed. Click **Clone** to create a new notification with the current values populated in the new notification. You can edit the values before saving.

Downloading Database.com WSDLs and Client Authentication Certificates

User Permissions Needed	
To download a WSDL:	“Customize Application”

You can download a Web Services Description Language (WSDL) document to integrate your applications with Database.com using the API.

The following WSDLs are available:

- Enterprise WSDL** - The enterprise WSDL is strongly typed, which means that it contains objects and fields with specific data types, such as `int` and `string`. Customers who use the enterprise WSDL document must download and re-consume it whenever their organization makes a change to its custom objects or fields or whenever they want to use a different version of the API.
- Partner WSDL** - Use this WSDL to build an integration that can work across multiple Database.com organizations, regardless of their custom objects or fields. Typically partners and ISVs use this WSDL. It is loosely typed, which means that you work with name-value pairs of field names and values instead of specific data types. The partner WSDL document only needs to be downloaded and consumed once per version of the API.
- Apex WSDL** - Use this WSDL to run or compile Apex in another environment. See the [Database.com Apex Code Developer's Guide](#) for details.
- Metadata WSDL** - Use this WSDL to migrate configuration changes between organizations or work with the customizations in your organization as XML metadata files. See the [Database.com Metadata API Developer's Guide](#) for details.

To download a WSDL document:

- Click **Develop > API**.

2. Download the appropriate WSDL by right-clicking the link for the appropriate WSDL document to save it to a local directory. In the right-click menu, Internet Explorer users can choose **Save Target As**, while Mozilla Firefox users can choose **Save Link As**.
3. On your computer, import the local copy of the WSDL document into your development environment.

Optionally, you can download a certificate to authenticate salesforce.com organizations. Use this certificate for workflow outbound messaging. This certificate is meant to identify that the request is coming from salesforce.com, not a specific user. If you want to use certificates to ensure secure connections using other Database.com features, such as Apex callouts, use [Database.com certificates and key pairs](#).

Click **Develop > API**, and on the **WSDL Download** page, right-click **Download Client Certificate** and save it to an appropriate location. You can then import the downloaded certificate into your application server, and configure your application server to request the client certificate.

Developing and Testing in a Test Database Organization

Managing Test Databases

Test Database Overview

User Permissions Needed	
To view a test database:	“View Setup and Configuration”
To create, refresh, activate, and delete test databases:	“Modify All Data”

Database.com gives you the ability to create multiple copies of your organization in separate environments for a variety of purposes, such as testing and training, without compromising the data and applications in your Database.com production organization. These copies are called test databases and are nearly identical to your Database.com production organization. For a list of differences, see [Test Database Setup Tips and Considerations](#) on page 433.

Test databases are completely isolated from your Database.com production organization, so operations you perform in your test databases do not affect your Database.com production organization, and vice versa.

The test database types are:

QA Database

QA databases are intended for coding and testing by a single developer. They provide an environment in which changes under active development can be isolated until they are ready to be shared. QA databases copy all application and configuration information to the test database. QA databases are limited to 10 MB of test or sample data, which is enough for many development and testing tasks. You can refresh a QA database once per day.

Staging Database

Staging databases copy your entire production organization and all its data, including custom object records. You can refresh a staging database every 29 days.

Test Database Limits

The following limits apply to test databases:

- You can refresh a staging database 29 days from its previous refresh or creation. If you delete a staging database, you must wait 29 days to replace it.
- You may order an unlimited number of QA databases.



Note: Contact salesforce.com to order test databases for your organization.

Test Database Retention Policy

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.

Managing Test Databases

User Permissions Needed	
To view a test database:	"View Setup and Configuration"
To create, refresh, activate, and delete test databases:	"Modify All Data"

To manage your test databases, click **Data Management > Test Database**. A list of your existing test databases displays.

- Click **New Test Database**.
- Database.com deactivates the **New Test Database** button when an organization reaches its test database limit. If necessary, contact salesforce.com to order more test databases for your organization.
- Note that Database.com deactivates all refresh links if you have exceeded your test database limit.
- Click **Show Test Database Refreshes** to see a log of your test database refresh history, including when test databases were created and who created them.
 - Click **Refresh** to replace an existing test database with a new copy. Database.com only displays the **Refresh** link for test databases that are eligible for refreshing. For staging databases, this is any time after 30 days from the previous creation or refresh of that test database. For QA databases, you can refresh once per day. Your existing copy of this test database remains available while you wait for the refresh to complete. The refreshed copy is inactive until you activate it.
 - Click **Activate** to activate a refreshed test database. You must activate your refreshed test database before you can access it. Database.com only displays this option for test databases that are not activated.



Caution: Activating a refreshed test database replaces the existing test database with the refreshed version. This permanently deletes the existing version and all data in it. Your production organization and its data will not be affected.

- Click **Del** to delete a test database. If you delete a staging database, you must wait 29 days before replacing it with a new staging database.



Caution: Deleting a test database permanently erases the test database and all data in it. Your production organization and its data will not be affected.

- Click **Login** to log in to a test database. Database.com only displays this option for active test databases.

Note that the **Login** button is for administrators and may not always be available; however, you can log into an active test database at <https://test.database.com> by entering your modified username and password. To change your username or password, see [Changing Your Password](#) on page 77.

- Click on a test database name to view details about the test database, including the test database type and when it was created.

Creating or Refreshing a Test Database

User Permissions Needed	
To view a test database:	"View Setup and Configuration"
To create, refresh, activate, and delete test databases:	"Modify All Data"

To create or refresh a test database:

- Click **Data Management > Test Database**.
- Do one of the following:

- Click **New Test Database**.

Database.com deactivates the **New Test Database** button when an organization reaches its test database limit. If necessary, contact [salesforce.com](#) to order more test databases for your organization.

Note that Database.com deactivates all refresh links if you have exceeded your test database limit.

- Click **Refresh** to replace an existing test database with a new copy. Database.com only displays the **Refresh** link for test databases that are eligible for refreshing. For staging databases, this is any time after 30 days from the previous creation or refresh of that test database. For **QA** databases, you can refresh once per day. Your existing copy of this test database remains available while you wait for the refresh to complete. The refreshed copy is inactive until you activate it.

- Enter a name and description for the test database. You can only change the name when you create or refresh a test database.



Tip: We recommend that you choose a name that:

- Reflects the purpose of this test database, such as "QA."
- Has few characters because Database.com automatically appends the test database name to usernames and email addresses on user records in the test database environment. Names with fewer characters make test database logins easier to type.

- Select the type of test database:

- QA Database:** QA databases are intended for coding and testing by a single developer. They provide an environment in which changes under active development can be isolated until they are ready to be shared. QA databases copy all

application and configuration information to the test database. QA databases are limited to 10 MB of test or sample data, which is enough for many development and testing tasks. You can refresh a QA database once per day.

- **Staging Database:** Staging databases copy your entire production organization and all its data, including custom object records. You can refresh a staging database every 29 days.



Note: Database.com enables you to create a QA database. To create a staging database, contact salesforce.com.

If you have reduced the number of test databases you purchased, but you still have more test databases of a specific type than allowed, you will be required to match your test databases to the number of test databases that you purchased. For example, if you have two staging databases but purchased only one, you cannot refresh your staging database as a staging database. Instead, you must choose one staging database to convert to a smaller test database, such as a QA database.

If you are refreshing an existing test database, the radio button usually preselects the test database type corresponding to the test database you are refreshing.

Whether refreshing an existing test database or creating a new one, some radio buttons may be disabled if you have already created the number of test databases of that test database type allowed for your organization.

5. Click **Start Copy**.

The process may take several minutes, hours, or even days, depending on the size of your organization.



Tip: You should try to limit changes in your production organization while the test database copy proceeds.

6. You will receive a notification email when your newly created or refreshed test database has completed copying. If you are creating a new test database, the newly created test database is now ready for use.

If you are refreshing an existing test database, an additional step is required to complete the test database copy process. The new test database must be activated. To delete your existing test database and activate the new one:

- a. Return to the test database list by logging into your production organization and navigating to **Data Management > Test Database**.
- b. Click the **Activate** link next to the test database you wish to activate.

This will take you to a page warning of removal of your existing test database.

- c. Read the warning carefully and if you agree to the removal, enter the acknowledgment text at the prompt and click the **Activate** button.

When the activation process is complete, you will receive a notification email.



Caution: Activating a replacement test database that was created using the **Refresh** link completely deletes the test database it is refreshing. All configuration and data in the prior test database copy will be lost, including any data changes you have made. Please read the warning carefully, and press the **Activate** link only if you have no further need for the contents of the test database copy currently in use. Your production organization and its data will not be affected.

7. Once your new test database is complete, or your refreshed test database is activated, you can click the link in the notification email to access your test database.

You can log into the test database at `test.database.com/login.jsp` by appending `.test_database_name` to your Database.com username. For example, if your username for your production organization is `user1@acme.com`, then your username for a test database named “test” is `user1@acme.com.test`.



Note: Database.com automatically changes test database usernames but does not change passwords.

Test Database Setup Tips and Considerations

User Permissions Needed	
To view a test database:	“View Setup and Configuration”
To create, refresh, activate, and delete test databases:	“Modify All Data”

Consider the following before you create a test database.

Servers and IDs

- The organization IDs of your test databases differ from your production organization ID, and will change each time your test database is refreshed.
- Database.com stores test database organizations on several instances. When a test database is created or refreshed, an instance is selected for your test database, so your test database may appear on different instances and have different URLs.
- When data that contains object IDs is copied from your production instance into your test database, the object IDs in your test database match the object IDs in your production instance. However, data created in your production instance or test database will not contain matching object IDs.

Username and Email Address Modification

- User information is included in a test database copy or refresh for all test database types. Because all Database.com usernames must be unique and reference a single organization, all copied usernames are modified to ensure uniqueness during the copy process. Usernames are modified differently for each test database copy. Entering a particular modified username will log you into a specific test database.
- For each username, the copy process applies one or two modifications as necessary to generate a unique new username:
 - First, the test database name is appended to the username, so that for a test database named `test`, `user@acme.com` may become `user@acme.com.test`.
 - If the resulting username is not unique, a second modification is performed in which a number of characters and digits are prepended to the modified username. This second modification may result in a username such as `00x7Vquser@acme.com.test`.
- Email addresses are modified in a test database so that production users, who may not know of the test database, do not receive automatically generated email messages from the test database. By modifying user email addresses, any email messages sent from the test database are not delivered to production users.
- You can manually correct email addresses in the test database user records for users who will use the test database for testing and training.



Caution: Test databases automatically change Database.com user email addresses. To avoid sending unsolicited email from your test databases, manually invalidate or delete all email addresses in your test databases that do not belong to users.

Creating, Refreshing, and Deleting Test Databases

- Test database copy is a long-running operation that occurs in the background. You are notified of the completion of a test database copy via email. Test database refreshes may complete in minutes, days, or even more than a week.
- A number of conditions factor into the duration of a test database copy or refresh, including the number of customizations, data size, numbers of objects (for full copies), and server load. Also, test database refreshes are queued, so your requested copy may not start immediately after your request.
- A test database is not a point-in-time snapshot of the exact state of your data. Furthermore, we recommend that you limit changes to your production organization while a test database is being created or refreshed. Setup and data changes to your production organization during the test database creation and refresh operations may result in inconsistencies in your test database. You may detect and correct some inconsistencies in your test database after it is copied or refreshed.
- Some types of test databases may not be available to choose from if you already reached your organization's limit of the types of test databases you can create or refresh. For example, if your organization is limited to one staging database, and a staging database is already created, you may not select **Staging Database** when creating a new test database. However, you may refresh your existing staging database.
- When you are finished with a test database, you can refresh it to create a new copy. However, if you have reduced your organization's number of test databases, a **Delete** link displays next to existing test databases, allowing you to delete a test database of your choice. Note that you must delete a test database before you can refresh any more test databases.

Accessing Test Databases

- Access changes for test database users:
 - ◊ A test database refresh deletes and recreates the test database as a new copy of the production organization. In effect, this reverses any manual access changes you've performed. If you created test database-only users, they will no longer exist, and a user's profile and permissions revert to their values in the production organization. This means that after a refresh, any access changes will need to be repeated in the new copy.
 - ◊ You can create users in your production organization that are inactive, and then activate them in a test database. This is a good way to create a user in a test database that has the appropriate permissions to develop in a test database. For more information, see [Editing Users](#) on page 196.
 - ◊ Many development and testing tasks require the "Modify All Data" permission. Because your developers might not have that permission in the production organization, you may need to increase their permissions in a test database. Exercise caution when granting this permission in test database organizations that contain sensitive information copied from production (for example, social security numbers). For more information on permissions, see [Overview of User Permissions and Access](#) on page 207.
 - ◊ You can create new users for test database development, but these count against the number of licensed users in your organization. To reduce your license count, you can disable production users who won't need access to the test database. For more information, see [Deactivating Users](#) on page 198.
 - ◊ To grant users access to a test database, you must log in as the administrator on the test database organization, and then create or upgrade user access in the test database.
- Always log in to your test database organization using the `https://test.database.com` login URL.
- Remember to log in using the modified username as described in [Username and Email Address Modification](#) on page 433.
- If using the API, after you log in you must use the redirect URL that is returned in the `loginResult` object for subsequent access. This URL reflects the instance on which the test database is located and the appropriate server pool for API access.
- All test database copies are made with federated authentication with SAML disabled. Any configuration information is preserved, except the value for `Recipient URL` changes to `http://tapp0.database.com`. The `Recipient URL` is updated to match your test database URL, for example `http://cs1.database.com`, after you re-enable SAML. To enable SAML in the test database copy, click **Security Controls > Single Sign-On Settings**; then click **Edit**, and select `SAML Enabled`. You must change the value of the `Recipient URL` in the certificate for your client application as well. For more information, see [Configuring SAML Settings for Single Sign-On](#) on page 278.

Test Database Storage Limits

- Staging databases have the same storage limit as your production organization.
- QA databases have a 10 MB storage limit.
- Test databases do not send email notifications when storage limits are reached. However, if you reach the storage limit of your test database, you cannot save new data in your test database. To check your storage limits, click **Data Management** > **Storage Usage** in your test database.

Customization and Data Changes

- Customizations and data changes in your production organization do not automatically appear in your test databases. You must create a new test database or refresh an existing one to see the customizations made to your organization since the last time you created or refreshed a test database.
- You can only add, edit, or delete Apex using the Database.com Console. In a Database.com production organization, you can only make changes to Apex by using the `compileAndTestAPI()` call. For more information, see the [Database.com Apex Code Developer's Guide](#).

Other Service Differences

- Database.com has a background process that permanently deletes records in the Recycle Bin that are older than 30 days. This process runs at different times on different servers, so its timestamp in your test database differs from its timestamp in your production organization. Applications and integrations that depend on this timestamp may fail if they are first connected to one environment, such as your production organization, and then later connected to another environment, such as your test database. Keep this in mind when developing applications and integrations that depend on this timestamp.

Note that the time of the latest execution of the background delete process is available through the `getDeleted()` API call. For more information, see the [Web Services API Developer's Guide](#).

Test Database Restrictions and Licenses

Test database services are restricted if your organization doesn't comply with salesforce.com's licensing rules. This typically happens when test database licenses expire.

Staging Database License

Permits the use of a staging or QA database.

QA Database License

Permits the use of a QA database only.

There are a few different types of restrictions you might encounter when your organization doesn't comply with licensing rules.

Unable to Refresh a Particular Type of Test Database

Cause—Your organization is using more test databases of a specific type than its test database licenses permit.

Example—Your organization has three staging databases, but only two staging database licenses.

Effect—You can't refresh a test database of this type. In the example, you can't refresh staging databases.

Resolution—Delete test databases to comply with the number allowed by your organization's test database licenses, or purchase more test database licenses.

All Test Databases of a Particular Type are Locked

Cause—The license count of a given type, including higher hierarchical types, is zero.

Example—Your organization has three staging databases and zero staging database licenses.

Effect—All test databases of a particular type are locked. You don't have access to the test databases.

Resolution—Purchase the correct test database licenses to unlock the test databases. If you don't purchase enough licenses, you can't refresh test databases of that type.

All Test Databases are Locked

Cause—Your production organization is locked.

Example—Your organization has one staging database and one QA database, but you can't log in to either test database.

Effect—If your production organization is locked, all test databases associated with the organization are locked.

Resolution—Contact your salesforce.com representative to unlock your organization. When your production organization is unlocked, the test databases are unlocked as well.

Migrating Changes Between Organizations

Deployment Overview

User Permissions Needed	
To edit deployment connections:	“Deploy Change Sets”
To use outbound change sets:	“Create and Upload Change Sets,” “Create AppExchange Packages,” AND “Upload AppExchange Packages”
To use inbound change sets:	“Deploy Change Sets”

Click **Deploy** to access tools used for migrating metadata changes between organizations.

Deployment Connections

In order to use the change sets feature, a deployment connection is required. You can specify connection permissions for both outbound and inbound change sets on the Deployment Connections page.

Outbound Change Sets

Make changes in the organization you are logged into, and upload those changes to another organization.

Inbound Change Sets

Accept, modify, or reject change sets uploaded from other organizations.

Monitor Deployments

Monitor the progress of deployments made through the Metadata API.



Note: Configuration changes deployed using change sets do not appear on the Monitor Deployments page.

Working with Change Sets

Change Sets Overview

User Permissions Needed	
To edit deployment connections:	“Deploy Change Sets”
To use outbound change sets:	“Create and Upload Change Sets”
To use inbound change sets:	“Deploy Change Sets”

A *change set* is a means by which one organization can send customizations to another organization. For example, you could create a new object in a test database organization and send it to your production organization using a change set. Change sets can only contain modifications you can make through the Setup menu; therefore, you can't use a change set to upload a list of widget records. In other words, change sets contain *metadata*, not data.

When you want to send customizations from your current organization to another organization, you create an *outbound change set*. Once you send the change set, the receiving organization sees it as an *inbound change set*.

Sending a change set between two organizations requires a deployment connection. Currently, change sets can only be sent between organizations that are affiliated with a production organization, for example, a production organization and a test database, or two test databases created from the same organization.

About Permission Sets and Profile Settings in Change Sets

Developers can use permission sets or profile settings to specify permissions and other access settings in a change set. When deciding whether to use permission sets, profile settings, or a combination of both, consider the similarities and differences.

Behavior	Permission Sets	Profile Settings
What permissions and settings are included?	<ul style="list-style-type: none"> • Custom object permissions • Custom field permissions • User permissions (such as “API Enabled”) • Apex class access 	<ul style="list-style-type: none"> • Custom object permissions • Custom field permissions • Apex class access
Added as a component?	Yes	No. Profiles are added in a separate setting.
Require supporting components to be installed?	For custom objects and fields, yes. For example, object permissions for the custom object <i>Items</i> are included only if the <i>Items</i> object is also included. However, user permissions don't require supporting components to be installed.	Yes

Components Available in Change Sets

The following types of components may be added to a change set.



Note: The components available for a change set vary by edition.

- Apex Class
- Apex Sharing Reason
- Apex Trigger
- Custom Data Type
- Custom Field
- Custom Label
- Custom Object
- Custom Object Criteria Sharing Rule
- Custom Object Owner Sharing Rule
- Custom Setting
- Group
- Language Translation
- List View
- Permission Set
- Queue
- Remote Site
- Role
- S-Control
- Validation Rule
- Workflow Field Update
- Workflow Outbound Message
- Workflow Rule



Note: If you create or modify components that aren't available in a change set, you can't send those components from one organization to another in a change set. In this case, migrate the changes manually by repeating the steps you performed when you created or modified the component.

Working With Deployment Connections

Deployment Connections

User Permissions Needed	
To edit deployment connections:	“Deploy Change Sets”

In order for change sets to be sent from one organization to another, a deployment connection is required between the organizations. Deployment connections can't be created between arbitrary organizations; instead, a deployment connection is created between all organizations affiliated with a production organization. For example, if you have a production organization (Prod) and two test databases (Dev and Test), a deployment connection is created between production and each test database (Prod and Dev, and another connection between Prod and Test), as well as between the test databases (Dev and Test).

A deployment connection alone doesn't enable change sets to be sent between organizations. Each organization must be authorized to send and receive change sets. This added level of security enforces code promotion paths and keeps organizations' setup metadata from being overwritten by mistake.

Viewing and Authorizing Deployment Connections

Viewing Available Deployment Connections

A deployment connection enables customizations to be copied from one organization to another. The deployment connections list shows which organizations are authorized to upload changes to this organization, and which organizations allow this organization to upload changes to them.

To view available connections, click **Deploy > Deployment Connections**.

Action

Click **Edit** next to the organization that you want to allow or disallow change sets from.

Name

A list of organizations that have deployment connections to the organization you are currently logged into. Click the name of an organization to view more information about the connection.

Description

A brief description of the connected organizations.

Type

The type of organization you are connected to. Possible values are Production, Staging Database, and QA Database.

Upload Authorization Direction

The arrows show the direction in which uploads can occur. A broken line means that no change sets are authorized in either direction. To authorize the connected organization to send you inbound change sets, edit the deployment connection for this organization. If you want to send outbound change sets to a connected organization, the administrator for that organization must edit the connection for that organization.

Viewing Details of a Deployment Connection

A deployment connection enables customizations to be copied from one organization to another. The deployment connections list shows which organizations are authorized to upload changes to this organization, and which organizations allow this organization to upload changes to them.

To view connection details:

1. Click **Deploy > Deployment Connections**.
2. Click the name of the organization you want to view.

Name

The name of the selected organization. This is not the organization you are logged into.

Description

A brief description of the organization.

Type

Possible values are Production, Staging Database, and QA Database.

Allow Inbound Changes

If selected, the named organization can send change sets to the organization you are currently logged into. This is a read-only field and can only be modified by selecting Allow Inbound Changes in the target organization.

Accepts Outbound Changes

If selected, the named organization allows change sets to be sent to it from the organization you are currently logged into.

Authorizing a Deployment Connection

In order for another organization to send change sets to the organization you are logged into, you must authorize the inbound change set:

1. Click **Deploy > Deployment Connections**.
2. Click **Edit** next to the organization you want to authorize.
3. Select **Allow Inbound Changes**.
4. Click **Save**.

Migrating Changes to Another Organization

Outbound Change Sets

User Permissions Needed	
To create, edit, or upload outbound change sets:	“Create and Upload Change Sets”



Watch a Demo (2:30 minutes)

An *outbound change set* is a change set created in the organization you are logged into and that you want to send to another organization. Typically, an outbound change set is used for customizations created and tested in a test database and then sent to a production organization.

Sending an outbound change set to another organization doesn't guarantee that the changes will be implemented in that organization. The change set must be deployed (accepted) by the target organization before the changes take effect.



Note: Change sets are limited to 2,500 components and a total file size of 400 MB.

Creating an Outbound Change Set

An outbound change set is a change you want to send from the organization you are logged into to another organization. To view outbound change sets, click **Deploy > Outbound Change Sets**.

- To create a new change set, click **New**.
- To view the details of an existing change set, click its name.

Viewing and Adding Dependent Components to a Change Set

A dependency is a relationship where one or more components must exist for another component to exist. It's a good idea to add dependent components to a change set, unless you are sure that the dependent components exist in every organization where this change set will be deployed.

To add dependent components to an outbound change set:

1. Click **Deploy > Outbound Change Sets**.
2. In the Change Sets list, click the name of a change set.
3. Click **View/Add Dependencies**.

- On the Component Dependencies page, select the dependent components you wish to deploy and click **Add to Change Set**.



Caution: If your change set contains more than 2500 dependencies you will only be able to see the first 2500 in the view dependencies page.

Selecting Components for an Outbound Change Set

To select the components in an outbound change set:

- Click **Deploy > Outbound Change Sets**.
- In the Change Sets list, click the name of a change set, or create a new one.
- Click **Add** to add components.
- Choose the type of component and the components you want to add, and then click **Add to Change Set**.
- Click **Add Profiles** to add profile settings to the change set.
- Optionally, click **View/Add Dependencies** to add dependent components.



Note: Dependent components rely on the existence of other components. Unless you are certain that the dependent components exist in every organization this change set will be deployed to, it's a good idea to add dependent components to the change set.

Cloning an Outbound Change Set

You can create a copy of an existing change set by cloning it.

- Click **Deploy > Outbound Change Sets**.
- Click the name of the change set you want to clone.
- Click **Clone**.

Uploading an Outbound Change Set

Once you've assembled the components in a change set, you can upload it to another organization. Note that once you upload a change set, you can't edit it or recall it.

- Click **Deploy > Outbound Change Sets**.
- Click the name of a change set.
- Select the organization you want to send the change set to.
- Click **Upload**.



Note: Outbound change sets expire six months after upload, at which time the change set is permanently deleted.

Deleting an Outbound Change Set

To delete an outbound change set:

- Click **Deploy > Outbound Change Sets**.
- Click the name of the change set you want to delete.
- Click **Delete**.

Viewing the Source of a Metadata Component

Before you upload or deploy a change set, you can view the source of each metadata component in the change set. This allows you to verify that you are uploading or deploying the correct changes.

If the metadata component in the change set is a profile or permission set, then the XML source displayed depends on the contents of the change set. For example, profiles only contain field-level security for fields included in the custom object that are part of the change set.



Note: Like profiles, permission sets contain permissions for only the custom components included in a change set. Unlike profiles, however, permission sets can also contain standard object permissions, standard field permissions, and user permissions (such as “API Enabled”).

For information about the fields, attributes, and settings of the available metadata types, see the [Database.com Metadata API Developers Guide](#)

Uploading Change Sets During Server Upgrades

During server upgrades, production and test database environments may not be running the same version of the platform. Some components may have new functionality or other changes that will not allow you to deploy that type of component until the production organization is running the same version as test database.

If you upload a change set that has components that can't be deployed, the system detects which components can't be deployed, and gives you the option of uploading the remaining components.

For example, Apex classes are automatically upgraded whenever a new version of the platform is released. If you create a change set in a test database during the test database preview and then upload that change set to a production organization, the system won't allow you to upload Apex classes. Likewise, other types of components that have changes that are incompatible between versions will be removed from the change set. However, components that remained unchanged between versions can be migrated freely.

Outbound Change Set Validation Errors

If you receive an error about cross-version validation, then the organization used to create the outbound change set is running on a different platform version than the organization receiving the change set. This error typically occurs during upgrades, because organizations may be upgraded at different times. If you receive this error, you can only deploy those components that are compatible between versions.

Importing Changes from Another Organization

Inbound Change Sets

User Permissions Needed	
To deploy inbound change sets:	“Deploy Change Sets”



[Watch a Demo](#) (2:30 minutes)

An *inbound change set* is a change set that has been sent from another organization to the organization you are logged into. A change sent must be *deployed* for the changes to take effect. You can deploy the contents of an inbound change set as a whole, but not on a component-by-component basis.

Viewing Inbound Change Sets

The Inbound Change Sets page lists change sets awaiting deployment, as well as the history of deployed change sets. To view inbound change sets, click **Deploy > Inbound Change Sets**.



Note: Inbound change sets are permanently deleted six months after the change set is uploaded.

Viewing Change Set Details

The Change Sets detail page lists information about a particular change set.

1. Click **Deploy > Inbound Change Sets**.
2. Click the name of a change set.

Validating a Change Set

You can validate a change set without deploying changes. Validating a change set allows you to view the success or failure messages you would receive with an actual deploy.

1. Click **Deploy > Inbound Change Sets**.
2. Click the name of a change set.
3. Click **Validate**.



Note: You can't make any changes to your organization while a test deployment is in progress.

4. After the validation completes, click **View Results**.

Deploying a Change Set

To deploy a change set:

1. Click **Deploy > Inbound Change Sets**.
2. In the Change Sets Awaiting Deployment list, click the name of the change set you want to deploy.
3. Click **Deploy**.

A change set is deployed in a single transaction. If the deployment is unable to complete for any reason, the entire transaction is rolled back. After a deployment completes successfully, all changes are committed to your organization and the change set can't be rolled back.



Note: Database.com requires that at least 75% of your code is covered by unit tests before you can deploy it to a production organization. Ideally, you should strive for 100% coverage. The code coverage restriction is not enforced for test databases.

Change Sets: Best Practices and Tips

Change Sets Implementation Tips

Authorization required to upload changes

Before you can deploy a change set from one organization to another, an administrator in the target organization must authorize uploads across the deployment connection between the two organizations.

Deployment Connections list displays all connections

The Deployment Connections list is automatically populated with your production organization and all test databases. It is possible to deploy between any of these organizations, but no other organizations.

Change set connections unavailable during maintenance

Authorizing deployment connections and uploading pages require information from the production organization, and are unavailable when production is undergoing maintenance. During this time you can construct outbound change sets but not upload them.

Test databases must be available

If an organization has no test databases provisioned, the user may see an Insufficient Privileges error on the Deployment Connections page.

Deployment is a one-way transaction

A change set is deployed in a single transaction. If the deployment is unable to complete for any reason, the entire transaction is rolled back. After a deployment completes successfully, all changes are committed to your organization and the change set can't be rolled back.

User fields are not preserved

Changes deployed in a change set or through the Metadata API, do not preserve user fields. These fields are either removed or replaced with the deploying user, depending on the context.

Change Sets Best Practices

Deploy all dependent components

Make sure each outbound change set contains all interdependent components that don't exist in the target organization. If you try to deploy a component that refers to another component missing from the target organization and from the change set, the deployment fails.

Change sets give you fine-grained control over what you deploy. For example, you can migrate custom fields individually. To deploy a custom object and all of its fields, you must add the custom object and every field to the change set; adding just the custom object to the change set won't cause deployment to fail, but results in an empty custom object.

Add permissions and access settings to outbound change sets

Adding profiles or permission sets to outbound change sets allows administrators to migrate permissions for users so they can access the new functionality. Profiles contain access settings for more components than permission sets, including page layouts, record types, and tab settings. However, permission sets contain access to standard object permissions, standard field permissions, and user permissions (such as "API Enabled").

Clone a change set to add dependent components to an uploaded change set

After you upload a change set, you can't change its contents. If you need to add dependent components to a change set you already uploaded, clone the change set, add the dependent components, and then upload it again.

Plan deployments around maintenance schedule

Plan your deployment activities around the maintenance schedule for both your production and test database organizations. Some features require information from your production organization when accessed from a test database.

Validate change sets before deployment

You can perform a test deployment of an inbound change set to view the success or failure messages that would occur with an actual deployment. This is a good idea if you are planning a deployment on a schedule (for example during low-use hours) and want to determine if the deployment will succeed ahead of time. However, you don't need to perform a test deployment every time you deploy, as this process takes time to complete. To test deploy an inbound change set, click its name and then click **Validate**.

View component details

You can view the XML representation of a component before you upload an outbound change set or deploy an inbound change set.

Change sets limited to 2500 components and 400 MB

Change sets are limited to 2500 components and a total file size of 400 MB. If your change set exceeds either of these limits, you can create separate change sets for email templates, dashboards, and reports. These components are often the most numerous and have fewer dependencies.

Deleting and renaming components

You can't use change sets to delete or rename components. To delete components, use the Web interface on the target organization. To rename a component, first delete the component on the target organization and then upload the new component in a change set.

Working with the Developer Console

Working with the Developer Console

User Permissions Needed	
To use the Developer Console:	"View All Data"
To use the execute anonymous text entry box:	"Author Apex"
To save changes to Apex classes and triggers:	"Author Apex"

The Developer Console is a collection of tools you can use to analyze and troubleshoot applications in your Database.com organization. It's a separate window composed of a set of related tools that allow you to access your source code and review how it executes. It can also be used to monitor database events, workflow, callouts, validation logic, cumulative resources used versus system limits, and other events that are recorded in debug logs. It's a context-sensitive execution viewer, showing the source of an operation, what triggered that operation, and what occurred afterward. Access the Developer Console by clicking **Your Name > Developer Console**.

To learn about the different sections of the Developer Console, see [Navigating within the Developer Console](#) on page 446.

To learn about the tools available in the Developer Console, see [Working with Views in the Developer Console](#) on page 452.

To learn more about some typical ways you might use the Developer Console, for example, evaluating Apex triggers, tracking DML, or monitoring performance, see [Examples of Using the Developer Console](#) on page 460.

What Can You Use the Developer Console For?

The Developer Console puts essential tools for editing code, debugging requests, and analyzing performance and memory usage in one place. You can use the Developer Console for a variety of administrative and development tasks, including:

General Debugging and Troubleshooting

The Developer Console provides mechanisms to inspect executed requests. It provides access to a fine-grained log, allowing you to review every statement executed within a request, and a set of interactive panels that let you simulate an execution "step-through."

The Developer Console provides a convenient set of tools to efficiently track down logical issues. For example, if you want to understand why a certain request generates an “Attempt to de-reference a null object” error, you can review the execution, identify the offending logic, and set a heap dump capture marker at that point. You can then execute the process again, and inspect the request at that specific point in the execution to understand in detail how to improve your code. While the Developer Console can't pause execution like a traditional debugger, it provides cloud developers much of the same visibility, and reduces the need to instrument code with `System.debug` commands. And unlike a traditional debugger, you can run the execution both forward and backwards.

Source Code Editing and Navigation

The Developer Console allows you to create and edit source code within the Developer Console itself, letting you inspect, trace, and edit your code all in the same tool. The Repository tab allows you to quickly browse through and open your source code. You can open a working set of source code views and switch between them with a single click. You can open and edit Apex triggers and classes, and you can open a read-only view of your object definitions.

Performance Validation

The Developer Console has a number of panels dedicated to the inspection of performance. You can open a debug log and review the Executed Units tab, which breaks up the request both by time and type. This tab categorizes the timings by methods, queries, workflows, callouts, DML, validations, and triggers, which gives you a clear idea of where to find performance issues. The Timeline tab provides a timeline view of the overall request. Within this view, you can review each of the timeline blocks, which interactively filters the corresponding log, allowing you to walk through the events for a given block. The Limits panel provides a summary view of resources used and maps them against your allocated request limits.

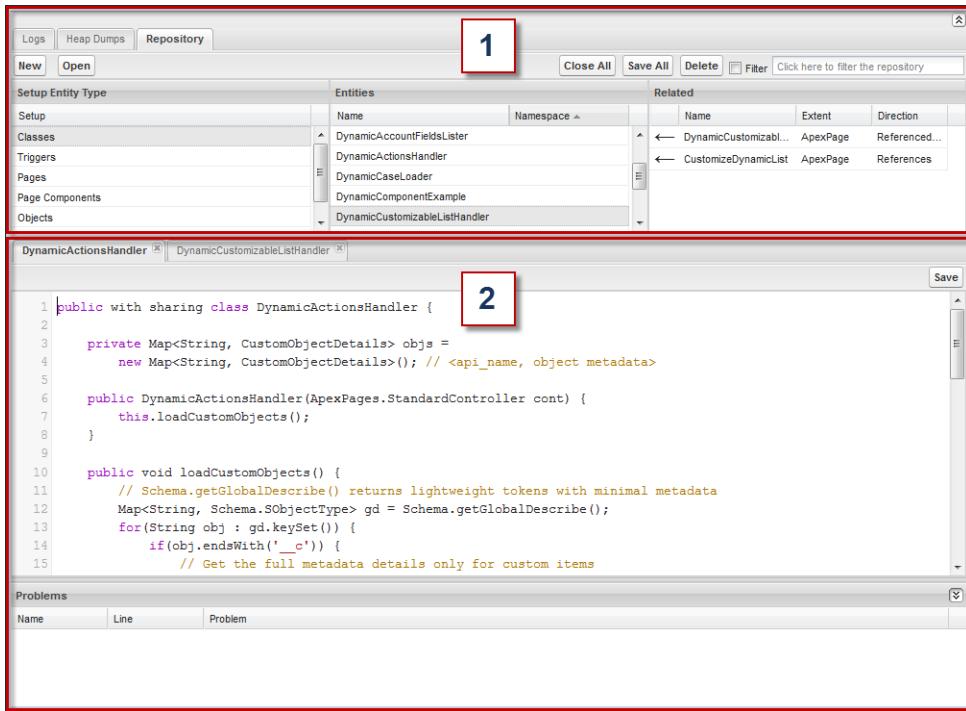
Understanding the Developer Console Interface

Navigating within the Developer Console

The Developer Console is a separate window composed of a set of related tools that allow you to access your source code and review how it executes. It can also be used to monitor database events, workflow, callouts, validation logic, cumulative resources used versus system limits, and other events that are recorded in debug logs.

The Developer Console is organized into two main sections:

1. A navigation panel for browsing to sources and developer logs.
2. A workspace where you open views to edit or examine items under investigation.



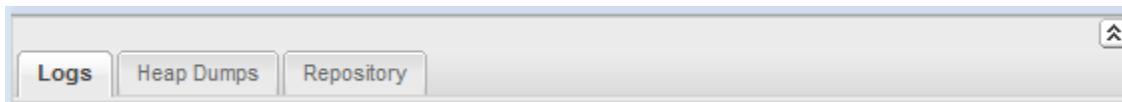
You can collapse the navigation panel and some subpanels of different detail views using the or button in the top right corner of the panel, to allow a larger viewing area for items being examined.

The workspace can display multiple log, code, and object definition views, each in its own tab. The arrangement of your tabs, and the panels within those tabs, including collapsed state of panels and your editing position in source files, is remembered when you close and re-open the Developer Console.

Navigation Tabs

The Developer Console has three tabbed navigation panels used for browsing to things you can examine:

- **Logs**—Allows you to open existing debug logs and, when the Developer Console is open, allows you to capture new logs from actions you take in your organization.
- **Heap Dumps**—Allows you to open heap dumps—a snapshot of the state of execution, that is, objects and variables, and references to and from instantiated objects—with links to the related source code.
- **Repository**—Allows you to navigate to and open your objects, Apex classes, triggers, and so on.



These tools allow you to open their respective logs, heap dumps, code, and object definitions. Each item opens in a new tab in your workspace, and you can have many tabs open at once. The arrangement of your tabs, and the panels within those tabs, including collapsed state of panels and your editing position in source files, is remembered when you close and re-open the Developer Console.



Note: A limited number of customers won't see the Heap Dumps tab. Contact salesforce.com if you want to examine heap dumps in your organization but don't see the Heap Dumps tab.

Viewing Debug Logs

Logs Tab

Use the Logs tab to browse to and open debug logs that include database events, Apex processing, workflow, callouts, and validation logic.

Open a log by double-clicking it. The log opens in a new System Log tab in the workspace. You can also open a log by selecting it and clicking **Open**.

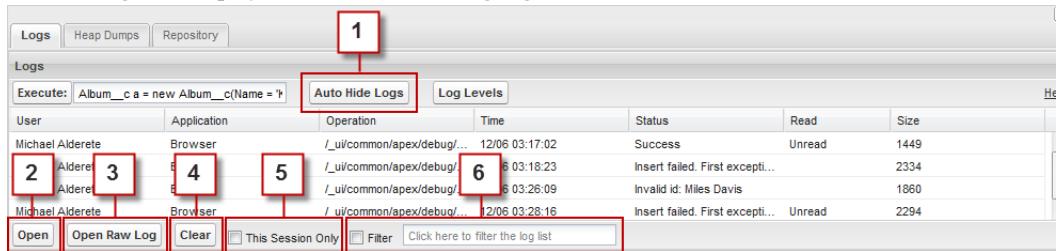
The Developer Console automatically polls for the current user's debug logs. After opening the Developer Console and clicking on the Logs tab, run something that generates a debug log. For example, if you have validation rules associated with inserting a new record, insert a new record. The Developer Console automatically captures a debug log for the request and adds it to the logs list.

 **Tip:** Logs are maintained in the Developer Console for 60 minutes or until you close the Developer Console.

To learn more about some typical ways you can use the System Log view to review a debug log, for example, evaluating Apex code, tracking DML, or monitoring performance, see [The System Log View](#) on page 452.

List of Existing Debug Logs

You can filter which logs are displayed in the list of debug logs.



User	Application	Operation	Time	Status	Read	Size
Michael Alderete	Browser	/ui/common/apex/debug/...	12/06 03:17:02	Success	Unread	1449
Michael Alderete	Browser	/ui/common/apex/debug/...	12/06 03:18:23	Insert failed. First excepti...		2334
Michael Alderete	Browser	/ui/common/apex/debug/...	12/06 03:26:09	Invalid Id: Miles Davis		1860
		/ui/common/apex/debug/...	12/06 03:28:16	Insert failed. First excepti...	Unread	2294

1. Click **Auto Hide Logs** to automatically hide all existing logs the next time the page is refreshed. This button is a toggle: click the button a second time to display all logs again.
2. Click **Open** to open the selected log in a new System Log view.
3. Click **Open Raw Log** to open the selected log in a plain text view that displays the unformatted contents of the debug log.
4. Click **Clear** to remove all logs from the list.



Tip: If you are monitoring debug logs for a user, those logs are still accessible from the Debug Log page. Click [Monitoring > Debug Logs](#).

5. Select **This Session Only** to display only logs generated by you since opening the Developer Console. Deselect to see all debug logs currently saved for your organization, including those created by monitoring users.
6. Filter the logs by clicking **Filter** and entering text. For example, if you only want to see a specific user's debug logs, filter by that user's name. The filter is case-sensitive.

To learn more about some typical ways you can use the System Log view to review a debug log, for example, evaluating Apex code, tracking DML, or monitoring performance, see [The System Log View](#) on page 452.

Setting Logging Levels

Click **Log Levels** to specify the logging levels for future requests. Logging levels determine how much information about a request is saved in a debug log.



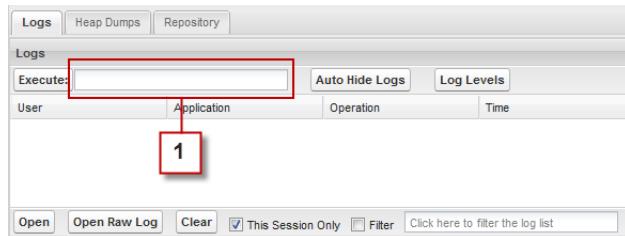
Tip: Larger logs take longer to load, so reducing the amount of logging may improve the responsiveness of the Developer Console.

If your logs are especially large you can customize the logging levels on a class-by-class basis by [setting debug log filters for your Apex classes and triggers](#). This will allow you to log fine-grained events for code you're reviewing, and coarse-grained events for code you're not interested in.

Execute Anonymous Text Entry Box

You can run brief Apex code snippets in the Developer Console to generate new debug logs covering specific application logic:

1. Click the **Execute** text entry box.



2. Enter your code in the popup window. For example:

```
Album__c a = new Album__c('Kind of Blue');
Insert a;
System.debug('The Album Id is: ' + a.id);
```

3. Click **Execute**.

Code that you enter into the execute text entry box runs as if you had executed it using `ExecuteAnonymous`. A new debug log with the results of the execution is added to the Logs list. Double-click the log to open a new System Log view with information about your executed code.

If you have already entered a snippet of code you want to re-test, click **Execute**. Code snippets are saved when you close and re-open the Developer Console.



Note: You can't use the keyword `static` in code that you enter directly into the Developer Console.



Caution: If you call a class that contains a `testMethod`, all DML statements of the test method are executed. This might add unwanted data to your organization.

Viewing Heap Dumps

Heap Dumps Tab

Use the Heap Dumps tab to browse and open heap dumps that preserve a snapshot of the state of objects in memory at the time of the capture. Open a heap dump by double-clicking it. The heap dump opens in a new Heap Dump Inspector tab in the detail area.

Heap Dumps				Heap Dump Locations			
Namespace	Class	Line	Time	File	Line	Iteration	
	ActorValidator	10	08/03 12:01:27		ActorValidator	14	1
	ActorValidator	10	08/03 12:01:49				
	ActorValidator	10	08/03 12:02:29				
	ActorValidator	13	08/03 12:04:06				
	ActorValidator	13	08/04 13:32:59				

Clear **This Session Only**

Capture new heap dumps by setting capture markers in your code. See [Enabling Heap Dump Captures in Apex Code](#).

To learn more about some typical ways you might use a Heap Dump Inspector tab, for example to evaluate why a branch of execution is or isn't being taken or to verify the expected contents of a variable, see [Heap Dump Inspector](#) on page 456.

Heap Dumps List

The Heap Dumps list contains the heap dumps currently available for review. Select **This Session Only** to only display heap dumps generated since opening the Developer Console. Deselect to display all heap dumps currently saved for your organization.

Each heap dump in the list displays this information:

Column	Description
Namespace	The namespace of the package containing source code marked to capture this heap dump.
Class	The Apex class containing the source code marked to capture this heap dump.
Line	The source line number marked to capture this heap dump.
Time	The time this heap dump was recorded.



Tip: Right click any column header to sort the information in the column. You can also select which columns you want displayed in the Heap Dumps list.

Heap Dump Locations

This is a list of the locations in source code where heap dumps are captured. Each item in the list displays this information:

Column	Description
File	The name of the Apex class that contains a marker to capture a heap dump.
Line	The line number marked to capture a heap dump.
Iteration	The number of times to execute the marked line of code before capturing the heap dump.

By default the iteration is 1, which means that the heap dump is captured the first time the line of source code executes. You can double-click the iteration number and change it, for example, to investigate why a loop does not terminate when expected. Only one heap dump will be captured for a specific line of code, no matter how many times it's executed during a request.

For details about how to set new heap dump capture markers in your source code, see [The Source Code View](#) on page 458.



Note: Heap dump locations are active while the Developer Console is open. Closing the Developer Console clears the locations.

Viewing Source Code and Object Definitions

Repository Tab

Use the Repository tab to browse to and open your application code and data objects. To navigate to an item:

1. In the **Setup Entity Type** column, click the type of the item you'd like to look at.
2. In the **Entities** column, scroll and find the item you'd like to examine.
3. Click the item once to see related items in the **Related** column. For example, click on an object to see the Apex classes that use it.
4. Double-click an item to open it in a new tab. You can also open an item by selecting it and clicking **Open**.



Note: Source code such as Apex classes and triggers, open in a Source Code editor, while custom objects open in a Schema Quick View.

Repository Management and Filtering

You can create new code such as Apex classes and triggers in the Repository browser. You can also open and close items, edit and save them, save changes across all tabs, and delete existing items.

Setup Entity Type	Entities	Related
Setup		
Classes	actorcon ActorValidator fluffy GISCalcAlt MyApex	← RoleValidation ApexTrigger Referenced by Actor CustomFieldDefinition References First Name CustomFieldDefinition References Last Name CustomFieldDefinition References Name CustomFieldDefinition References
Triggers		
Objects		

1. **New**—Creates a new Apex class or trigger, depending on which entity type you have selected in the Setup list. To create a new Apex trigger, first select the object on which to create the trigger.
2. **Open**—Opens the selected item, an alternative to double-clicking.
3. **Close All**—Closes all tabs open in the detail area. If any tab contains unsaved changes, you'll be prompted to save them.
4. **Save All**—Saves changes in all tabs open in your workspace. This allows you to [save a set of dependent changes](#) on page 459.
5. **Delete**—Deletes the selected item. You can only delete Apex classes and triggers.

6. You can filter which items are displayed in the Repository browser. Click the **Filter** text entry box and enter a text string to display only items that match the filter criteria. The search is case-sensitive.



Note: You can't modify custom objects in the Developer Console. To create, edit, or delete custom objects, click **Create > Objects** to work with your custom objects.

Working with Views in the Developer Console

Working with Views in the Developer Console

You can open multiple tabs in the Developer Console workspace. You can open as many tabs as your screen will allow. The arrangement of your tabs, and the panels within those tabs, including collapsed state of panels and your editing position in source files, is remembered when you close and re-open the Developer Console.

You can collapse the top, bottom, and many sub-panels using the , , , or button in the top right corner of the panel. This is especially useful when working with the information-rich System Log and Source Code views.

When collapsed, you can click on a panel to temporarily reveal and use it. When your cursor moves out of the panel, it collapses automatically.

You can open four different views in the workspace:

- [System Log view](#)
- [Heap Dump Inspector view](#)
- [Source Code view](#)
- [Schema Quick View](#)

Using the System Log View

The System Log View

The System Log view is a context-sensitive execution viewer that shows the source of an operation, what triggered that operation, and what occurred afterward. You can use this view to inspect debug logs that include database events, Apex processing, workflow, and validation logic.

The System Log view contains these sections:

1. [Stack](#)
2. [Back trace](#)
3. [Execution log](#)
4. [Execution log filter controls](#)
5. [Source window](#)
6. [Variables](#)
7. [Executed Units tab](#)
8. [Limits tab](#)
9. [Timeline tab](#)

The screenshot shows the System Log view with several sections highlighted by red boxes and numbered 1 through 9:

- Stack:** Shows a tree structure of debug log operations. A folder labeled "anonymous_apex_fnc" is highlighted with box 1.
- Scop Unit Duration Heap:** A table showing resource usage for different operations. An "execi execute" row is highlighted with box 2.
- Execution Log:** A table of events with columns: Timestamp, Event, and Details. An event row is highlighted with box 3.
- Source:** A code editor showing a line of Apex code: "Account a = new Account(name='Acme')". A line in the code is highlighted with box 5.
- Variables:** A table showing variables with columns: Variable, Value, and Name. A variable entry is highlighted with box 6.
- Executed Units:** A table showing performance metrics for various operations. A row for "Method debug" is highlighted with box 7. A row for "Method execute" is highlighted with box 8. A row for "Method valueOf" is highlighted with box 9.
- Show:** A menu bar at the bottom with options: Methods, Queries, Workflow, Callouts, DML, Validations, Triggers, and Pages.

Many sections of the System Log view refresh automatically and display related information when you click on an item. For example, if you click a folder labeled handleProductPriceChange in the stack section, these sections are updated to display information about that trigger:

- Back trace
- Execution log
- Source

Similarly, if you click a line in the execution log, the stack, back trace, and source are all updated. Clicking items in the Executed Units tab also updates the execution log, stack, back trace, and source sections.

Stack and Back Trace

The stack section displays a tree structure of the debug log, starting with the top level operation. It displays information “top down”—from initiating calls to the next level down, and so on. Use this to see the hierarchy of items as they are called in the process. For example, if a class calls a second class, the second class is displayed as a child node of the first class.

The back trace (the panel below the stack) contains a “bottom up” view of the currently selected item in the debug log, starting with the lowest level call, followed by the operation that triggered that call, and so on. Each item in the back trace has this information:

Column	Description
Scope	Delimited region within the process, such as workflow, a class, or DML.
Unit	Name of the item (region).
Duration	Amount of time (in milliseconds) the item took to run.
Heap	Amount of heap (in bytes) the item used.

Execution Log

The execution log contains the [debug log](#) for the process. The debug log contains every action that occurred in the process, such as method calls, workflow rules, and DML operations. Hover over long lines in the section to display a popup of the entire line.

Timestamp	Event	Details
20:11:27:632	EXECUTION_STARTED	
20:11:27:632	CODE_UNIT_STARTED	[EXTERNAL]
20:11:28:240	VARIABLE_SCOPE_BEGIN	[1]a Accou
20:11:28:437	STATEMENT_EXECUTE	[1][1]
20:11:28:437	STATEMENT_EXECUTE	[1][1]
20:11:28:458	HEAP_ALLOCATE	[1]Bytes:4
20:11:28:460	HEAP_ALLOCATE	[1]Bytes:4
20:11:28:552	1 VARIABLE_ASSIGNMENT	[1]this.Narr
20:11:28:557	2 VARIABLE_ASSIGNMENT	[1]a [{"Name":
20:11:28:557	STATEMENT_EXECUTE	[1][1]

1 This Frame **2 Executable** **3 Filter** Click here to filter the log

Use the execution log to retrace the steps through a process. You can step through lines on your own. You can also use one of the following to filter the debug log to lines of specific interest:

- This Frame**—Displays only this region of the process. For example, if you click CODE_UNIT_STARTED and **This Frame**, the execution log displays only the items in the process that occur between CODE_UNIT_STARTED and its associated CODE_UNIT_ENDED. This displays only the items that are associated with that level. For example, if you have a trigger that calls a class, only the trigger operations are displayed in the execution log.
- Executable**—Displays only the executable items in the debug log. This hides the cumulative limits information, such as the number of SOQL queries made, the number of DML rows, and so on.



Tip: View the execution log with **Executable** checked. Only deselect it when you are working on optimizing your process and need the limits information.

- Filter**—Displays items that match what you enter in the **Filter** field. For example, if you type DML, all the lines in the execution log with the string DML in either the event or details are displayed. The filter is case-sensitive.

You can also click **Download** to download a copy of the debug log as a text file. The default name for the file is apex.log.

The execution log panel contains this information:

Column	Description
Timestamp	System time when the process began. This is shown in the local user's time. The format is: HH:MM:SS:MS.
Event	Debug event .
Details	Additional details pertaining to the event, such as line number and parameters.

Source

The source panel contains the executed source code or the metadata definitions of entities used during the process. What's displayed in the source section depends on what's selected elsewhere in the System Log view. The source section also lists how many times a line of code was executed.

Enter a line number in the entry box at the bottom of the source panel and click **Jump** to go to a specific line of code.

When viewing executed source code, click **Open** to open it in a new Source Code tab.



Note: If validation rules or workflow is executed during the process, the metadata representation of it displays in the source panel. You can't open a metadata representation from the Developer Console. See [ValidationRule](#) and [Workflow](#) in the [Database.com Metadata API Developers Guide](#).

Variables

Use the variables section to discover when a variable is assigned a value and what that value is. Click on a Variable event to populate the section.



Note: The **Apex Code** log level must be set to **Finest** for variable assignments to be logged.

Another way to view the contents of variables is to use the [Heap Dump Inspector](#), which allows you to see more details about entities held in memory at a point of execution.

Executed Units, Limits and Timeline

The Executed Units, Limits, and Timeline tabs appear in the Execution Overview panel at the bottom of the Developer Console when you are looking at a System Log view. The Execution Overview panel can be collapsed using the  button in the top right of the panel.

Use the Executed Units tab to examine which items in the process used the most system resources. Use the Limits tab to compare the resources used by the process to overall system limits. The limits are listed by name and amount. Use the Timeline tab to see a visual representation of the time each process took. Use this to identify and isolate the parts of your request that take the longest.

To filter out the information displayed on the Executed Limits tab by the type of item, click the button for that type. For example, to no longer view methods in the section, click **Methods**. Click the button a second time to display methods again.

Right-click any column header to sort the information in the column. You can also select which columns you want displayed in the Executed Units tab.

The **Executed Units** tab contains this information:

Column	Description
What	Type of process item. Types include: <ul style="list-style-type: none"> • Method • Queries • Workflow • Callouts • DML • Validations • Triggers

Column	Description
Name	Name of the process item.
Sum	Total duration for the item.
Avg	Average duration for the item.
Min	Minimum amount of time for the item.
Max	Maximum amount of time for the item.
Cnt	Number of times the item was called during the process.
Heap	Amount of space the item took on the heap.
Query Type	Type of query. Possible values are: <ul style="list-style-type: none"> • SOQL • SOSL
Sum rows	Total number of records changed for that item.
Avg rows	Average number of records changed for that item.
Max rows	Maximum number of records changed for that item.
Min rows	Minimum number of records changed for that item.

On the **Limits** tab, select the unit of time used to display the process. The default is Minutes.

The **Limits** tab contains this information:

Column	Description
Limit	Name of the limit.
Used so far	The amount of the limit used by this process at this point of execution.
Request total	The amount of this limit used by the request at completion.
Total available	The total amount for the limit.

The **Timeline** tab contains this information:

Column	Description
Category	The type of process.
Mills	Milliseconds of time taken by that process.
%	The percent this process took of the entire request.

Using the Heap Dump Inspector

The Heap Dump Inspector

The Heap Dump Inspector view lets you browse snapshots of the state of objects in memory at the time of the capture, including references between objects. It also lets you view variables in more detail than offered in the System Log view, including

individual items in collections. Use the Heap Dump Inspector to investigate what objects are in memory at a specific point of execution and why, that is, what other objects hold references to them.



Note: A limited number of customers won't see the Heap Dumps tab or be able to open Heap Dump Inspector views. Contact salesforce.com if you want to examine heap dumps in your organization but don't see the Heap Dumps tab.

Open existing heap dumps from the [Heap Dumps tab](#). Capture new heap dumps by setting capture markers in your code. See [Enabling Heap Dump Captures in Apex Code](#).

The Heap Dump Inspector lets you view a heap dump two different ways, in the Heap tab and the Symbols tab.

The Heap Tab

The Heap tab displays all objects in memory at the time of the heap dump capture. Items are listed and grouped by data type.

Types		Instances		State		
Type	Count	Total Size	Address	Size	Field	Value
Contact	1	28	0x229e70da	180	0	0x611d6744
DynamicObjectHandler	1	12			1	0x35b1d5cd
LIST<String>	1	180			2	0x6748d0c1
MAP<String, Schema.SObjectField>	1	196			3	0x59c43715
Schema.SObjectField	48	0			4	0x5d8f6109
String	48	537			5	0x49cae665
					6	0x6258a515
					7	0x757d6e5

References		Search	
Inbound References		Referencing Instances	
Field	Type	Address	Size
accessibleFields	DynamicObjectHandler	0x2ba10de0	12

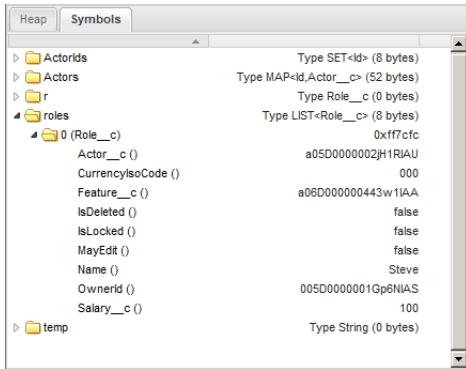
The **Types** column is a flat list of the classes of all instantiated objects in memory at the time of capture, with a count of how many are instantiated, and the amount of memory consumed in bytes. Click an item to see a list of those objects in the **Instances** column, with their address in the heap and memory consumed. Click an instance to view the variables currently set in that object in the **State** column.

The **References** tab provides two lists to display relationships between symbols held in memory. Use the **Inbound References** list to locate the symbols which can hold references to objects of a particular type. Use the **Referencing Instances** list to find specific instances holding references to a symbol. Double click to find that instance elsewhere in the heap.

The **Search** tab lets you find symbols in the heap by value or address. Search matches partial symbol values, but addresses must be exact. To quickly search for a value, click the search icon (🔍) that appears to the right of it when you hover over it in the State panel.

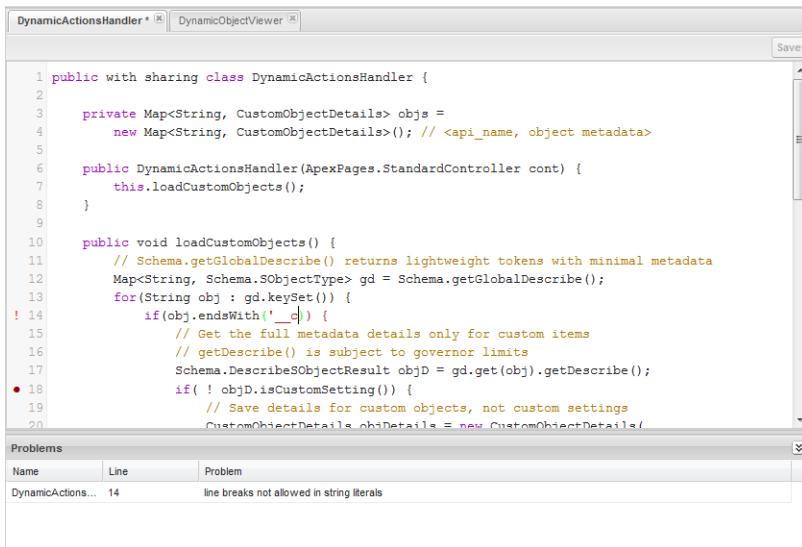
The Symbols Tab

The Symbols tab displays a tree view of all symbols in memory at the time of the heap dump capture. Use it to quickly review the state of the system at the specific line of code (and iteration) where the heap dump was captured.



The Source Code View

The Source Code view displays Apex source files and lets you edit them in the Developer Console. Using the Repository browser, you can open a set of related source code files and switch between them more quickly than using the sidebar.



When you make changes in a source code view, the name of the tab will have a “*” added to it to indicate there are unsaved changes. Click **Save** in the corner of a source view to save changes to it.

 **Note:** You can't save if you have compilation errors. [Review the Problems panel](#), correct any errors, and click **Save** again.

Click **Save All** in the Repository browser to save changes in all open tabs. When you save changes using **Save All** all modified views are saved together in one request. This allows you to [save a collection of changes with dependencies](#), which would otherwise be blocked because separately they would be invalid.

Enabling Heap Dump Captures in Apex Code

To set a marker for a new heap dump capture point, click in the margin to the left of the line numbers. A heap dump will be captured immediately *before* executing that line. By default the heap dump will be captured the first time the line of code is

executed. You can change the iteration for the capture in the [Heap Dump Locations](#) list. Only one heap dump will be captured for a specific line of code, no matter how many times it's executed during a request.



Note: Heap dump locations are active while the Developer Console is open. Closing the Developer Console clears the locations.

You can set heap dump capture markers in Apex classes and triggers.



Note: If you set a capture marker in a method with the `@future` annotation, to capture that heap dump during a request you need to keep the Developer Console open until the `@future` method completes asynchronously. This is because heap capture markers are reset when you close the Developer Console.

Validating Changes in Source Code

Changes you make in a Source Code view are validated in the background. While you are editing code an error indicator displays on lines causing errors, and the Problems panel shows the details of compilation errors. You can collapse the Problems panel using the button in the corner of the panel.

When validating your 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 adding a method in one file, and a call to that method in another, and other common development tasks—will not be reported as errors.

Using Save All to Save Changes with Dependencies

When you save modified source views they are validated against all other saved sources and, if there are errors, saving is prevented. It's common in the normal course of development to make changes to one source file that are dependent on related changes in another file. If the dependencies are mutual—a common occurrence when doing development—it then becomes impossible to save any of the related files individually.

The Developer Console allows you to save changes with dependencies across multiple unsaved files. Click **Save All** in the Repository browser to save changes in all open tabs. This will save all modified views together in one request. This allows you to save a collection of changes with dependencies, which would otherwise be blocked because separately they would be invalid.

Staying in Sync with Code in the Cloud

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.

The Schema Quick View

The Schema Quick View gives you a read-only reference for the fields of a custom object, and their data types.

Name	Apex Type
Id	Id
OwnerId	Id
IsDeleted	Boolean
Name	String
CreatedDate	Datetime
CreatedById	Id
LastModifiedDate	Datetime
LastModifiedById	Id
SystemModstamp	Datetime
Title__c	String
Author__c	String
ISBN__c	String
Price__c	Decimal
Publisher__c	String



Note: You can't modify custom objects in the Developer Console. To create, edit, or delete custom objects, click **Create > Objects** to work with your custom objects.

Using the Developer Console to Solve Problems

Examples of Using the Developer Console

Here are some of the ways you can use the Developer Console to diagnose and solve problems.

- Tracing the Path of Execution
- Viewing System.Debug Statements
- Updating Source Code
- Tracking DML in a Request
- Viewing a Complex Process

Tracing the Path of Execution

Scenario: You've opened a debug log in a System Log view. What are some of the ways to step through the information?

1. In the execution log, click **Executable**. This filters out all non-executable steps, including cumulative limits information.

Execution Log		
Timestamp	Event	Details
15:33:34:045	CODE_UNIT_STARTED	[EXTERNAL]execute_anonymous_apex
15:33:34:065	STATEMENT_EXECUTE	[1]
15:33:34:065	STATEMENT_EXECUTE	[1]
15:33:34:066	VARIABLE_ASSIGNMENT	[1][this.Name]"Test Account"0xaf25fe
15:33:34:066	VARIABLE_ASSIGNMENT	[1]a["Name":"Test Account"]0xaf25fe
15:33:34:066	CODE_UNIT_FINISHED	execute_anonymous_apex

This Frame
 Executable
 Filter
 Click here to filter the log

2. Click the Executed Units tab to view the aggregate values of different types of operations in the request. For example, you can view the number of DML operations or the different methods by the type of method.
3. Click the Limits tab to view the governor limits used by this operation.

Viewing System.Debug Statements

Scenario: You've added a number of System.Debug statements to your code to track a request's progress. How do you find them using the System Log view?

1. Click **Filter** in the execution log panel.
2. Enter debug in the entry box.

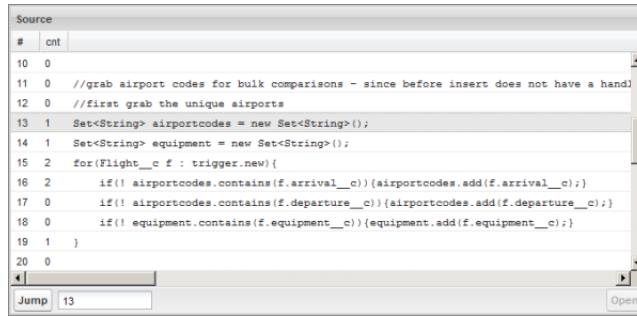
Only the lines containing the word `debug` in your request display.

Updating Source Code

Scenario: After you run your request, you notice an Apex code error in the debug log. What's the easiest way to edit your Apex code?

1. From the Source panel, select the line of code.
2. Click **Open**.

The Apex editor opens the source code for that class or trigger in a new Source Code tab.



```

Source
# cnt
10 0
11 0 //grab airport codes for bulk comparisons - since before insert does not have a handle
12 0 //first grab the unique airports
13 1 Set<String> airportcodes = new Set<String>();
14 1 Set<String> equipment = new Set<String>();
15 2 for(Flight__c f : trigger.new){
16 2     if(! airportcodes.contains(f.arrival__c)) {airportcodes.add(f.arrival__c);}
17 0     if(! airportcodes.contains(f.departure__c)) {airportcodes.add(f.departure__c);}
18 0     if(! equipment.contains(f.equipment__c)) {equipment.add(f.equipment__c);}
19 1 }
20 0
Jump 13 Open

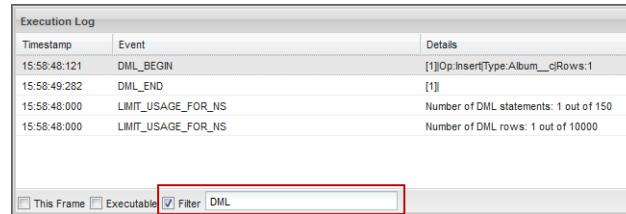
```

Tracking DML in a Request

Scenario: Suppose your request contains many DML statements in different locations. How can you tell how many times DML is executed in a request?

Here are two techniques for drilling into a debug log to examine the actual DML executed during the course of a request:

1. In the execution log, click **Filter**, then enter DML. All items in the request that contain DML anywhere in either the event or details display.



Execution Log		
Timestamp	Event	Details
15:58:48:121	DML_BEGIN	[1]Op:Insert Type:Album__c Rows:1
15:58:49:282	DML_END	[1]
15:58:48:000	LIMIT_USAGE_FOR_NS	Number of DML statements: 1 out of 150
15:58:48:000	LIMIT_USAGE_FOR_NS	Number of DML rows: 1 out of 10000

2. In the Executed Units tab, disable all other types of execution, except for DML. The buttons are toggles—click once to filter that type of operation **out** of the list. Click again to disable the filter. To view only the DML, click **Methods**, **Queries**, **Workflow**, **Callouts**, **Validations**, **Triggers** and **Pages**.

- The details of the DML operation show the kind of object that was affected, and the specific operation performed—insert, update, and so on. You can also view the number of times a DML statement was executed, the number of rows, and so on.
- If you click a DML request item in the Executed Units tab, the execution log section filters out all other parts of the request, and displays only that DML statement.

You can also use these procedures for looking up and filtering queries.

Viewing a Complex Process

Scenario: Your process is complex, and includes several Apex classes and triggers, workflow, and validation rules. What are some of the best ways to step through or filter the resulting debug log?

1. The stack section contains a tree structure illustrating the execution path of all the top level items in the request. Use this to see the hierarchy of items as they execute.
2. Use the **Filter** entry box in the execution log. For example, if you're interested in trigger-specific events, click **Filter** and enter `trigger`.

Timestamp	Event	Details
21:07:08:374	CODE_UNIT_STARTED	[EXTERNAL]01qD00000004Lif!DeleteRestrictAlbums on Album trigger event BeforeDelete for [a00D000...
21:07:08:769	CODE_UNIT_FINISHED	DeleteRestrictAlbums on Album trigger event BeforeDelete for [a00D0000008oLqb]

Execution Log

This Frame Executable Filter trigger

Only the lines in the debug log that contain the word `trigger` display in the execution log section.

3. Limit the scope of the execution log to a specific selected unit of execution by selecting **This Frame**. For example, if you select a line that contains `CODE_UNIT_STARTED` in the execution log, and then click **This Frame**, the execution log displays only the items in the request that occur between `CODE_UNIT_STARTED` and its associated `CODE_UNIT_ENDED`.



Note: When **This Frame** is selected the execution log only displays the items that are contained in that frame, *not* any lower level operations. For example, if a trigger calls a class, only the trigger operations display in the execution log, not the class operations.

Index

A

Access 207, 224
 about 207
 revoking 224
 Activating 73, 416
 critical updates 73
 workflow rules 416
 Activities 251
 controlled by parent 251
 Adding 195
 users 195
 Administrative permissions 225
 Ant task for Apex 407
 Apex 51, 58–59, 380–389, 391, 393, 403–406, 428, 445, 458, 460
 adding users to monitor 403
 class summary 383
 classes 381
 code 380
 creating a class 382
 creating a class from a WSDL 382
 debug log details 391
 debug log filters 393
 debug log levels 393
 debug logs 389
 debugging 445, 460
 defining a trigger 387
 dependencies 383
 downloading a custom WSDL 428
 editor 458
 job queue 51
 managing triggers 387
 monitoring system logs 58
 overview 380
 resetting debug logs 403
 setting class access 385–386
 setting class security 384
 source code 458
 test results 405
 testing 383, 403
 tests 404, 406
 trigger detail page 388
 version settings 386
 viewing a class 383
 viewing debug logs 59
 Apex Data Loader 347
 See Data Loader 347
 API 426, 428
 downloading a WSDL 428
 API usage 427–428
 details 428
 notifications 427
 App permissions 225
 Auditing 107, 242
 fields 107, 242

B

Background jobs 63
 about 63
 viewing 63
 Batch jobs 408

Browsers 46–48

Firefox settings 47
 Internet Explorer settings 48
 settings 46
 supported versions 46
 Bulk API 351, 359, 426
 Bulk data load jobs 53, 55
 monitoring 53
 viewing job details 55

C

Certificates 295–297, 299
 CA-signed 297
 creating 296
 editing 299
 uploading 297
 viewing 299
 Change sets 437–444
 best practices 444
 cloning 441
 deleting 441
 dependent components 440
 deploying 443
 deployment 442
 deployment connections 438–440
 details 443
 implementation tips 443
 inbound 442
 outbound 440–442
 permission sets and profiles 437
 permissions 437
 selecting components 441
 testing before deployment 443
 version errors 442
 viewing component source 442
 Chatter 410–412
 customizing feed tracking 411
 enabling 411
 setting up 411
 stop automatically following records 412
 Checkout 203
 granting access to users 203
 Classes 401
 debug logs 401
 Client authentication certificates 428
 downloading 428
 Cloning a change set 441
 Code 409
 security 409
 Command line 360–363
 configuration file (Data Loader) 362
 encrypted password (Data Loader) 361
 encryption key (Data Loader) 360
 field mapping file (Data Loader) 362
 importing data (Data Loader) 363
 introduction (Data Loader) 360
 prerequisites (Data Loader) 360
 Company information 2, 5
 editing 2
 fields 5
 language setting 2

Concurrent usage limits 246
 Configuring 414–415
 workflow actions 415
 workflow rules 414
 Cookies 255
 Copy 429
 organization 429
 Corporate currency 11
 See *Currency* 11
 Creating 80, 86, 91, 260, 414–415
 custom object relationships 91
 custom objects 86
 custom views 80
 groups 260
 workflow rules 414–415
 Critical updates 73
 activating 73
 overview 73
 Cross-object field updates 417
 Cross-object formulas 117
 Currency 6, 9–11
 active 6, 10
 conversion rates 10
 corporate currency 11
 inactive 6, 10
 multicurrency 6
 multicurrency setup 9
 supported 11
 using multiple currencies 9
 Custom fields 98, 101, 113, 117, 194
 attributes 101
 building formulas 117
 changing data type 113
 default values 98
 rich text area 194
 Custom objects 85–86, 88–91, 93–94, 99, 230, 243
 creating 86
 creating relationships 99
 delegated administration 243
 deleting 88
 deploying 85
 erasing permanently 89
 managing 85
 many-to-many relationships 93
 permissions 230
 related lists 99
 relationship considerations 94
 relationships 91
 restoring 89
 security 90
 undeleting 89
 Custom settings 67–72, 99
 accessing 68
 adding data 69
 adding fields 70
 creating fields 99
 defining 70
 hierarchy 67
 limits 71
 list 67
 managing 67
 managing data 72
 overview 67
 setting access levels 69
 understanding 67
 viewing 71
 viewing data 72

Custom summary formulas 132
 operators and functions 132
 Custom views 80–82, 209, 215
 creating and saving 80
 deleting 80
 field filters 81
 navigating 81
 permission sets 209
 profiles 215
 sorting 81
 special picklist values 81
 values for date fields 82
 Customizing 186, 232, 413
 picklist entries 186
 tags 232
 workflow 413

D

Data Loader 347–351, 354, 357, 359–367, 369–370
 attachments 359
 batch files 364
 batch mode parameters 370
 Bulk API 351, 359
 column mapping 369
 command line interface 366
 command line introduction 360
 command line operations 367
 config.properties 370
 configuration file (command line) 362
 configuring 351, 359
 configuring batch processes 366
 data types 348
 date formats 348
 encrypted password (command line) 361
 encryption key (command line) 360
 field mapping file (command line) 362
 importing data (command line) 363
 installed files 364
 installing 350
 overview 347
 password encryption 365
 prerequisites (command line) 360
 sample files 364
 settings 359
 Spring Framework 350
 starting batch processes 370
 system requirements 350
 third-party licenses 349
 troubleshooting 357
 uninstalling 354
 when to use 347

Data storage 63

Dates 82
 using in filter criteria 82

Deactivating 198
 users 198

Debug logs 58–59, 389, 391, 393, 401, 403
 adding users to monitor 403
 classes and triggers 401
 details 391
 filters 393
 levels 393
 monitoring 58
 removing users from monitoring 403
 resetting 403
 retaining 58, 403
 viewing 59

Debugging 58–59, 391, 393, 401, 403, 454–455
 adding users to monitor 403
 class and trigger log levels 401
 debug log details 391
 filtering 393
 level of logging 393
 monitoring logs 58
 profiling information 455
 removing users from monitoring 403
 resetting debug logs 403
 stepping through a process 454
 viewing logs 59
 workflow 455

Debugging Apex 445, 460

Default field values 98, 115
 about 115
 considerations 115
 defining 98

Defer sharing calculations 252

Defining 416, 419
 field updates 416
 outbound messages 419

Delegated authentication 293–294
 configuring single sign-on 294
 single sign-on 293

Deleted fields 106
 restoring 106
 undeleting 106

Deleting 80, 198
 users 198
 views 80

Deleting a change set 441

Dependencies 383

Dependent components 440

Dependent lookups 92, 101

Dependent picklists 188–189
 about 188
 defining 188
 editing 189
 limits 189

Deploying 406–408
 using Change Sets 406
 using the Force.com IDE 407
 using the Force.com Migration Tool 407
 using the Web Services API 408

Deployment monitoring 64

Developer Console 445–446, 448–449, 451–452, 456, 458–460
 about 445, 460
 Apex code 451
 Apex triggers 451
 code editor 458
 database 459
 debug logs 448
 debugging 449, 456
 developer logs 448
 heap dump 449, 456
 Heap tab 456
 Symbols tab 456
 layout 446
 logs 448
 memory 456
 navigation 446
 object 459
 organization 446
 schema 459
 sections 446
 source code 458
 symbols 456

Developer Console (*continued*)
 table 459
 tabs 452
 understanding 445, 460
 user interface 452
 variables 456
 views 452

Development 409
 security 409

Domain name 50
 rolling out 50
 setting up 50
 tips for implementing 50

E

Editing 74, 105, 186, 196, 260
 fields 105
 groups 260
 personal information 74
 picklist entries 186
 users 196

Enhanced profile user interface 49, 219–220
 about 219
 apps 220
 enabling 49
 system 220

Error page 270
 customizing in SAML 270

F

Feeds 411–412
 customizing Chatter feed tracking 411
 stop automatically following records 412

Field updates 416, 418
 considerations 418
 defining 416

Field-level security 231, 238, 263–265
 accessibility 263
 permission sets 231, 265
 profiles 231, 265

Fields 5, 74, 98–99, 101, 105, 107, 109, 175, 190, 192–193, 207, 242, 251, 263–265
 access 264
 accessibility 263
 adding 99
 auditing 107, 242
 company information 5
 creating 99
 custom field attributes 101
 default values 98
 editing 105
 field-level security 264
 history 107, 242
 managing 109
 permissions 265
 roles 207
 roll-up summaries 192
 roll-up summary 190
 sharing model 251
 tracking changes 107, 242
 universally required 193
 user 74
 validation rules 175

File storage 63

Filtering debug logs 393

Force.com IDE 426
 Force.com Migration Tool 426
 Formulas 98, 116–117, 124, 132, 175
 about 116
 building 117
 cross-object 117
 default values 98
 examples 124
 operators and functions 132
 validation rules 175

G

General permissions 225
 Generating security keys 296
 Group membership calculations 254
 Groups 260–261
 about 260
 creating and editing 260
 member types 260
 viewing all users 261
 viewing lists 261

H

History 107, 242
 fields 107, 242

I

Identity provider 267, 300, 304–305, 308
 about 300
 editing 304
 enabling 304
 example 308
 values 267
 viewing details 305
 Identity providers 307–308
 error log 307
 examples 308
 Identity URLs 282
 Inbound change set 438, 440, 442
 Inline editing 210, 216
 permission sets 210
 profiles 216
 Integration 428
 downloading a client authentication certificate 428
 downloading a WSDL 428
 IP addresses 217, 255
 trusted 217, 255
 whitelist 217, 255

J

Job Queue for Apex 51
 Junction objects 94
 considerations 94
 Just-in-time provisioning 271
 example SAML assertions 271
 Just-in-Time provisioning 286
 requirements 286
 Just-in-Time provisioning errors 288

K

Key pairs 295–296
 creating 296

L

Languages 2
 setting the organization language 2
 Licenses 197
 Database.com users 197
 Portal 197
 viewing by type 197
 List views 80
 See Custom views 80
 Lists 81
 navigating 81
 sorting 81
 Locale 20
 supported 20
 Logging in 202, 270
 as another user 202
 SAML start page 270
 Logging out 270
 SAML 270
 Login 65, 217, 221–224, 240–241, 255, 257–259, 300, 304
 activation 217, 255
 enabling identity provider 304
 failures 65
 history 65
 hours, restricting 217, 221, 223, 255, 259
 identity confirmation 217, 255
 identity provider 300
 IP address ranges, restricting 217, 222, 224, 255, 257–258
 restricting 257
 restricting IP addresses organization-wide 241
 service provider 300
 session security 240
 trusted IP addresses 217, 255
 Lookup filters 92, 101
 dependent lookups 92, 101
 Lookups 92, 101
 dependent 92, 101

M

Mail merge 109
 fields 109
 overview 109
 Manual sharing 239
 Many-to-many relationships 93–94
 considerations 94
 creating 93
 Master encryption keys 295, 298
 Merge fields 109
 guidelines 109
 syntax 109
 Metadata API 64, 426
 deployments 64
 Metadata components in change sets 437
 Modify All permission 230–231
 Monitoring 53, 55, 421–422
 bulk data load job details 55
 bulk data load jobs 53
 notifications for outbound message queue 421
 outbound message queue 421
 outbound message queue notification requests 421

Monitoring (*continued*)
 outbound message queue notifications 421
 viewing outbound message queue notification requests 422
 Monitoring metadata deployments 64
 Multicurrency 6, 9
 See Currency 6, 9
 My Domain 49
 overview 49

N

Network access 241

O

OAuth 282, 314–316, 318, 320–321, 326–328, 333, 336, 339, 341–342, 345
 authenticating 320
 authentication flow 321, 328, 333, 336, 339
 defining remote access applications 318
 endpoints 321
 error codes 326
 JWT Web token flow 341
 refresh token flow 336
 revoking tokens 327
 SAML assertion flow 345
 SAML Bearer flow 342
 selecting a version 316
 terminology 315
 user-agent authentication flow 333
 username-password authentication flow 339
 using access token 342
 using identity URLs 282
 version 1.0.A authentication flow 321
 Web server authentication flow 328
 Object permissions 230–231
 Object-level security 238
 OpenID Connect 282
 Organization 429
 copy 429
 Organization profile 2
 See Company information 2
 Organization-wide sharing settings 238, 246, 248, 251
 about 238
 specifying 246, 248
 Outbound change set 438, 440, 442
 Outbound change sets 440–442
 selecting components 440–441
 version errors 442
 Outbound messages 419, 421–422
 defining 419
 notification requests 421
 notifications 421
 outbound message queue 421
 port restrictions 419
 tracking delivery status 421
 viewing notification requests 422

P

Passwords 78, 198–199, 201–202, 255
 changing by administrator 201
 changing by user 78, 198
 expiring 255
 expiring all passwords 202
 policies 255
 settings and controls 199

Permission sets 208–213, 225, 230, 238, 265
 about 208
 app permissions 225
 apps 211
 assigning 213
 cloning 209
 creating 209
 deleting 211
 editing 210
 field permissions 265
 list views, creating and editing 209
 navigating 212
 object permissions 230
 object security 238
 overview page 211
 searching 212
 system 211
 system permissions 225
 viewing 211
 Permissions 207, 220, 224–225, 230–231, 265
 about 207
 administrative 225
 app 225
 field 231, 265
 general 225
 Modify All 230
 object 230–231
 revoking 224
 searching 220
 system 225
 user 225
 View All 230
 Personal tags 232
 deleting for deactivated users 232
 enabling 232
 Picklist dependencies 189
 editing 189
 Picklists 186–188
 adding or editing entries 186
 dependent 188
 replacing values 187
 sorting 186
 Portals 246
 organization-wide defaults 246
 Profiles 213–217, 219–225, 230, 238, 257–259, 264–265, 442
 about 213
 assigned users 217
 cloning 217
 creating 217
 creating list views 215
 deleting 214, 219, 222
 editing 216
 editing, original user interface 223
 enhanced list views 214
 enhanced user interface, about 219
 field permissions 265
 field-level security 264
 login hours 221, 223, 259
 login IP address ranges 222, 224, 257–258
 object permissions 230
 object-level security 238
 overview page 219
 searching 220
 user permissions 225
 viewing 219, 222
 viewing in a change set 442
 viewing lists 214
 Public groups 260

Public tags 232
enabling 232

Q

Queues 233–234
cases 233
custom objects 233
deleting 234
editing 234
knowledge article versions 233
leads 233
managing 233
overview 233
service contracts 233
setting up 234
viewing 233
viewing members 234

R

Records 412
stop automatically following 412
Relationships 91, 93–94, 99, 106
adding 99
considerations 94
defining 99
many-to-many 93
overview 91
restoring 106
undeleting 106
Remote access 282, 314–321, 327–328, 333, 336, 339, 341–342, 345
authenticating users 320
authentication flow 321, 328, 333, 336, 339, 345
defining applications 318
delete 317
deny access 320
details 317
developing for 320
JWT Web token flow 341
managing applications 317
OAuth 336
 scope 336
overview 314
request approved 319
requests 319
revoking access 327
SAML Bearer flow 342
scope 336
starting 316
terminology 315
using access token 342
using identity URLs 282
Remote site configuration 313
Replacing picklist values 187
Rich Text Area fields 194
Role hierarchies 238
 about 238
Roles 203, 205–207
 assigning to users 205
 assigning users to 205
 editing 205
 fields 207
 managing 203
 sharing groups 206
 viewing 203
 viewing user lists 206

Roll-up summaries 192
 defining 192
Rotating master encryption keys 295, 298
Rules, sharing 239
 See Sharing rules 239
Running Apex test 404, 406

S

SAML 266, 270–271, 277–280, 286, 288, 300, 304, 345
 about 266
 custom error page 270
 enabling identity provider 304
 example assertions 271
 identity provider 300
 Just-in-Time provisioning 286
 Just-in-Time provisioning errors 288
 Just-in-Time provisioning requirements 286
 login history 277
 login page 270
 logout page 270
 OAuth 345
 prerequisites 266
 SAML assertion flow 345
 service provider 300
 single sign-on 278
 start page 270
 validating single sign-on 280
 validation errors 280
 viewing single sign-on 279
Scheduled jobs 53
 about 53
 viewing 53
Scheduling Apex 408
Searching 212, 220
 permission sets 212
 profiles 220
Security 90, 217, 222, 235–239, 241, 249, 255, 257–258, 264–265,
 286, 295, 298, 300, 304, 316, 409
 auditing 239
 browsers 237
 CAPTCHA 316
 certificates 295
 code 409
 cookies 255
 custom objects 90
 enabling identity provider 304
 field-level 238
 field-level security 264–265
 identity provider 300
 identity providers overview 304
 infrastructure 237
 Just-in-Time provisioning 286
 Just-in-Time provisioning requirements 286
 key pair 295
 login challenge 257
 login IP address ranges 222, 258
 login restrictions 217, 255
 manual sharing 239
 master encryption keys 295, 298
 network 257
 object-level 238
 organization-wide sharing settings 238
 overview 235
 queues 249
 record-level security 238
 restricting IP addresses organization-wide 241
 role hierarchies 238

- Security (*continued*)
 - service provider 300
 - session 239
 - sharing rules 239
 - single sign-on 255
 - SSL 239
 - timeout 239
 - token 217, 255
 - trust 236
 - user 255
 - user authentication 255
- Security and sharing 237
 - managing 237
- Security token 79
 - resetting by user 79
- Selecting 422
 - workflow actions 422
- Service provider 300, 306–308
 - about 300
 - defining 306
 - example 308
 - viewing details 307
- Service providers 305, 307–308
 - enabling 307
 - examples 308
 - mapping users 307
 - prerequisites 305
- Session security 240
- Setup 2, 73–74, 84, 243, 312
 - delegating setup tasks 243
 - menus, expanding and collapsing 84
 - monitoring changes 312
 - organization 2
 - personal information 74
 - searching 84
 - user 73
- Sharing 246–251
 - Apex managed 249
 - organization-wide defaults 246, 248
 - organization-wide sharing settings 249, 251
 - overrides 247, 249
 - rule considerations 250
 - rules, See Sharing rules 250
 - settings 246, 248–249
- Sharing groups 206, 260
 - See Groups 260
- Sharing model 231
 - object permissions and 231
- Sharing rules 239, 247, 249–254
 - about 250
 - categories 250
 - custom objects 251–252
 - defer sharing calculations 252
 - group membership calculations 254
 - notes 250
 - sharing rule recalculation 253–254
- Sharing, manual 239
 - See Manual sharing 239
- single sign-on 255, 267
 - identity provider values 267
- Single sign-on 266, 271, 277–280, 289, 292–295, 308, 345
 - best practices 292
 - configuring delegated authentication 294
 - debugging 280
 - delegated authentication 293
 - example 308
 - example SAML assertions 271
 - login errors 295
- Single sign-on (*continued*)
 - login history 277
 - OAuth 345
 - overview 289
 - prerequisites 266
 - SAML 278
 - SAML assertion flow 345
 - SAML validation 280
 - viewing 279
- Site 313
 - configuring remote 313
- Social APIs 410
- Spring Framework, see Data Loader 350
- Storage limits 63
 - data storage limits 63
 - file storage limits 63
- Support 79
 - granting login access 79
- System Log view 452–455
 - back trace 453
 - executed units 455
 - execution log 454
 - profiling information 455
 - sections 452
 - source section 455
 - stack 453
- System log, see Debug logs 403
- System permissions 225

T

- Tab Bar Organizer 49
 - enabling 49
- Tags 232
 - customizing 232
 - deleting for deactivated users 232
 - enabling 232
- Test database 429–431, 433, 435
 - creating 431
 - implementation tips 433
 - licenses 435
 - managing 430
 - refreshing 431
 - restrictions 435
- Testing 383, 403, 405
 - Apex test results 405
- Testing Apex 404, 406
- Time Zone 17
 - supported 17
- Time zone setting 74
- tokens, revoking 327
- Training history 66
- Transactions, replaying 445, 460
- Transferring 96
 - multiple records 96
 - records 96
- Transferring records 96
 - overview 96
- Triggers 387–388, 401
 - debug logs 401
 - defining 387
 - detail page 388
 - managing 387
- trust 236

U

Unit tests 404, 406
 Universally required fields 100, 193
 about 193
 considerations 100, 193
 Updates 73
 activating 73
 critical updates 73
 Usage limits 246
 concurrent 246
 User permissions 225
 User profiles 213
 See Profiles 213
 User roles 203
 See Roles 203
 User setup 74, 78–79, 198, 243, 260
 activating computer 78
 changing passwords 78, 198
 delegated administration 243
 editing 74
 fields 74
 granting login access 79
 groups 260
 public groups 260
 resetting security token 79
 Users 74, 194–198, 203, 205, 207, 213, 217, 224–225, 230
 access 207
 adding 195
 assigned to profiles 217
 assigning permission sets 213
 assigning roles 205
 changing profiles 196
 Database.com licenses 197
 deactivating 198
 deleting 198
 editing 196
 license types 197
 managing 194–195, 203
 object permissions 230
 permissions 207, 225
 revoking access 224
 revoking permissions 224
 See also User setup 74
 unlocking 196

V

Validation rules 172–176, 389
 about 172

Validation rules (continued)

- activating 174
- cloning 174
- considerations 173
- creating 175
- debug logs 389
- defining 175
- deleting 174
- examples 176
- useful formulas 176
- viewing 174

Version settings 386

View All permission 230–231

Viewing 206, 261

- all users in group 261
- users in role lists 206

Viewing Apex test results 405

Views 80

- See Custom views 80

W

Workflow 58, 62, 413–419, 421–423, 425

- actions 415, 422
- activating a rule 416
- considerations 423
- cross-object field updates 417
- debugging 425
- field updates 416, 418
- field updates that re-evaluate workflow rules 418
- immediate actions 425
- monitoring debug logs 58
- notifications for outbound message queue 421
- outbound message queue 421
- outbound message queue notification requests 421
- outbound message queue notifications 421
- outbound messages 419
- overview 413
- queue 62
- rules 414–416, 425
- terminology 413
- time triggers 423, 425
- time-dependent actions 423, 425
- tracking outbound messages 421
- viewing outbound message queue notification requests 422

Workflow metadata 455

Workflow rules 389

- debug logs 389

WSDLs 428

- downloading 428