

What's New in 4.0 ("Titan")

- Multi-tenancy: Multiple organizations with one implementation; dynamically updated WSDL for each org.
- Internet-facing deployment (IFD); CRM Online (hosted)
- Major workflow improvements via Workflow Foundation
- Reporting Wizard
- Duplicate detection; Bulk delete
- Async operations (plug-ins; workflow)
- Algorithm-based e-mail tracking (no token required)
- Data Migration Manager and Import Data Wizard
- Many-to-many and self-referential entity relationships
- Multilanguage; multicurrency

Selling Points

- CRM: Sales, Service, Marketing, Custom Entities; Free 90-day trial available
- Own or rent: On-premise, partner, or CRM Online
- Flexible access: Outlook (on/offline), browser, mobile, portal (SharePoint), custom UI
- Customization: Custom entities, Web/Outlook forms, JScript, Plugins, Workflow
- Web-based administration; Role-based security; Line of Business (LOB) platform
- Engineered for change (auto-adapting metadata-driven design); Office integration (Outlook, Excel reports and pivot tables, Word mail merge)

Major Product Components

- Web Application / Web Services
- Windows Workflow Foundation
- Data Connector for Reporting Services
- Asynchronous Service
- Discovery Service
- Deployment Service
- Metadata Repository
- Outlook Client (Add-in)
- E-mail Router (POP3/SMTP/Exchange)
- Data Migration Manager
- Duplicate detection
- Queues

CRM 4.0 Server Editions

- Workgroup: Maximum of five User licenses. Windows Small Business Server.
- Professional: Single tenant deployment. Multi-server. No user license limit.
- Enterprise: Multi-tenant, multi-server. No user license limit.

Deployment Scenarios

- All-in-One (Small Business)
- Team / Department (Medium Business)
- Division / Enterprise
- CRM Online
- Partner or Enterprise Hosted

On-premise Requirements

- Active Directory
 - Windows Server SBS/2003/2008
 - Internet Information Services (IIS)
 - SQL Server 2005 or 2008
 - SQL Server Reporting Services
 - Internet Explorer
 - Outlook
 - Visual Studio 2008
 - .NET Framework 3.0
- ## Optional Configuration
- Exchange 2003 or 2007
 - Microsoft Office SharePoint Server
 - SQL Analysis Services
 - Network Load Balancing
 - SQL Clustering
 - ISA Server
 - BizTalk
 - System Management Server
 - Operations Manager
 - System Center

Importing Data

- Import Data Wizard: Basic data importing; Start with delimited file; One file per entity; Assigned to single owner; 4 MB max file size; Automatic or manual data mapping; Admin UI shows success/failures
- Data Migration Manager: Import multiple source files at one time; Assign records to multiple users; Complex data mapping

Unsupported Customizations

- Rule: If not in SDK then consider it unsupported.
- Examples: Changing any CRM file (js, aspx, etc.), Adding custom files in folders other than Bin and ISV, Changing the database schema, Changing CRM website settings, Manipulating HTML DOM, Changing undocumented registry entries, Hide system views, Using custom HTTP modules, Creating virtual directory under CRM app, etc.

Integration

- UI (IFrame, Web Services, AJAX, Mashups)
- Plug-Ins
- SharePoint (Web Parts)
- EAI
- ETL

UI Customization (ISV.Config and SiteMap)

- SiteMap: Modify the application-level navigation pane and Workplace profiles; Set order of nav links
- ISV.Config: Add custom buttons, menus and entity-level navigation.
 - To pass parameters to the target URL, set PassParams attribute to true. Parameters: typename, type, id, orgname, userIcid, orgIcid
- Note: To edit SiteMap or ISV.Config, export to XML file, make changes, and import.

Miscellaneous

- CRM server tracing: Enable via the Registry; Trace data is written to Drive:\Program Files\Microsoft Dynamics CRM\Trace (See Implementation Guide)
- Access CRM database through SQL filtered views.
- MetaData browser: http://server[:port]/orgname/sdk/list.aspx
- Workflow or Plug-in? Use workflow when possible and async is ok; Use plug-in for synchronous processing, offline use, and data validation; If manual execution is needed then use workflow
- Dev tools: Search online for crmdiagtool4, "ie developer toolbar", fiddler, ieinspector, sysinternals, stunnware, "crm codeplex"

Performance and Scalability

- Service grid architecture
- Deployment choice
- WAN performance optimization
- Load balance clustering
- Web farm clustering
- Exchange clustering
- SQL Server clustering
- SQL mirroring
- Scale to: 24,000+ users
 - > 1M web requests / hour

Workflow

- CRM workflow runs as a Windows Service; provides asynchronous processing
- Can be initiated manually, automatically from triggering event, or from another workflow process
- User context: Manually started = context of the user who started it; Rule-based = workflow owner
- Trigger when record is Created, Status Changes, Assigned, Attributes Change, Deleted
- Actions: Create/Update/Assign record, Send e-mail, Start child workflow, Change status, Stop workflow, Custom actions

Outlook Client

- Supports Connected and Disconnected (Offline) modes
- Near identical functionality as web client
- Offline: Data copied to local MS SQL 2005 Express
 - Can schedule background interval data updates
 - Synchronize offline changes back to the server
 - Select data to download with Local Data Groups
 - Limitations: no workflow, async plug-ins, import
- Special feature: Activity reminders
- No Outlook Web Access (OWA) integration
- Built-in rules handle deleted data in Outlook and web
- Diagnostics tool provided for troubleshooting

Core System Entities

- Salesforce Automation: Account, Contact, Lead, Opportunity, Competitor, Quote, Order, Invoice, Product
- Marketing: Campaign, List, Literature
- Service: Case, Contract, KB Article
- Activities: E-mail, Fax, Letter, Phone Call, Task, Appointment, Service Activity, Campaign Response, Campaign Activity, Order Close, Quote Close, Opportunity Close, Quick Campaign, Case Resolution, System Job, Bulk Operation Log
- Other: Subject, Role, Team, Business Unit

Reporting Options

- Entity views and Advanced Find
- Dynamic Excel files
- SQL Server Reporting Services
- Reporting Wizard (Web UI)
- Filtered (SQL) views
- MS Word, Excel, Access
- Links to Web pages
- Third-party reporting tools

E-mail Functionality

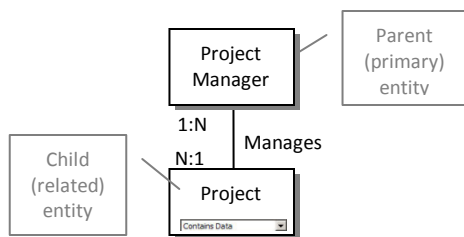
- Router: Interface between e-mail system (Exchange/SMTP/POP3) and Dynamics CRM; Outlook is an alternative on client computers
- Tracking: Automatic tracking of inbound/outbound e-mail; Optional tracking tokens; Smart matching; Individual-level settings
- Templates: Insert into messages; Bulk (direct) send; Use in workflows and system job notification; Supports data fields, ownership levels. Single entity or global. No use in Quick Campaigns.

Asynchronous Service

- Managed Queue: Executes asynchronous registered plug-ins, workflows, bulk mail, bulk import, bulk delete, and campaign activity propagation.
- "System Job" in UI = asyncoperation (SDK Class)
- Start/stop the service in Windows Services tool .
- Common uses for async plug-ins: recurring events, auditing, bulk operations, process-intensive operations.

Forms and Controls

- **Form Properties:** IsDirty; FormType (0=undefined, 1=create, 2=update, 3=readonly, 4=disabled, 5=quick create, 6=bulk edit); Objectid; ObjectTypeCode; ObjectTypeName; all.{field_name}
 - Syntax: `crmForm.propertyname`
- **Form Events:** onLoad; onSave (Refer to event.Mode property for type of save occurring)
 - To cancel save: `event.returnValue = false`
- **Form Methods:** Save, SaveAndClose; Save example: `crmForm.Save()`
- **Field Properties (all fields):** DataValue, Disabled, RequiredLevel, IsDirty, ForceSubmit
- **Field Events (all fields):** onChange (occurs when the data in a form field has changed and focus is lost)
- **Field Methods (all fields):** SetFocus, FireOnChange
- **Field Types:** Boolean, Customer, Date/Time, Duration, E-mail Body, Float, Integer, Lookup, Memo, Money, PartyList, Picklist, Regarding, State, StatusReason, String
- **Picklists:** A picklist contains multiple options. Properties and methods: DataValue (get/set; get returns string of selected option; set changes the selected option); SelectedText (get; returns string value of selected option); GetSelectedOption (returns a picklist option object); Options (get/set; array of Picklist option objects); AddOption; DeleteOption; DefaultValue (get/set default value)
- **Date/Time:** Can be formatted to show both the date and time, or the date only. Properties: DataValue (get/set), TimeVisible (get only).
- **Regarding:** A field that represents a Lookup where the linked record can be any record type that can be presented in the field. Supports the DataValue property.
- **Customer:** A field that represents a type of Lookup where either an account or contact is a valid type of record.



Entity Relationships

- **Relationship Types:** one-to-many, many-to-one, many-to-many, self-referential.
- **Behavior Types:** Parental, Referential, Referential / Restrict Delete, Cascade All, Cascade Active, Cascade User-Owned, Cascade None
- **Cascading Rules:** Assign, Share, Unshare, Reparent, Delete, Merge
- **N:N limitations:** No added attributes; "Add Existing Record" limits search to 100 records; No workflow support

SDK

Web services:

- **Discovery:** `http://CrmServer/mscrmservices/2007/ad/crmdiscovereservice.aspx`
 - Used to find correct CrmService endpoint for an organization; obtain ticket
- **CRM:** `http://CrmServer/mscrmservices/2007/crmservice.aspx`
 - Strongly-typed access to all entities; Methods: Create, Retrieve, RetrieveMultiple, Update, Delete, Fetch, Execute
- **Metadata:** `http://CrmServer/mscrmservices/2007/metadataservice.aspx`
 - Read/write metadata (entity/attribute/relationship definitions)

Assemblies:

- **Microsoft.Crm.Sdk.dll:** Contains methods needed for developing plug-ins and custom workflows. Use `DynamicEntity` class. Method: `Execute`.
- **Microsoft.Crm.SdkTypeProxy.dll:** Contains types needed for developing plug-ins and custom workflows.

Accessing CrmService (Sample Code)

```
public CrmService GetCrmService(string orgName, string server) {
    CrmAuthenticationToken token = new CrmAuthenticationToken();
    token.AuthenticationToken = 0 // Active Directory (On-Premise);
    token.OrganizationName = orgName;
    CrmService service = new CrmService();
    service.Credentials = System.Net.CredentialCache.DefaultCredentials;
    service.CrmAuthenticationTokenValue = token;
    service.Url =
        string.Format("http://{0}/mscrmservices/2007/crmservice.aspx", server);
    return service;
}
```

Plug-ins (Plugins)

Write plug-ins using any .NET 2.0 CLR-compliant language. Inherit from `IPlugin`. Sign (strong name) each assembly. Handle pre-event and post-event.

Basic plug-in class example:

```
using System;
using Microsoft.Crm.Sdk;
using Microsoft.Crm.SdkTypeProxy;
namespace MyPlugins
{
    public class AccountCreateHandler: IPlugin {
        public void Execute(IPluginExecutionContext context) {
            try {
                // Do something here. See SDK for samples.
            } catch (SoapException ex) {
                throw new InvalidPluginExecutionException("err",ex)
            }
        }
    }
}
```

// Note: If plug-ins consist of multiple assemblies, consider using `ILMerge`

Events (Messages):

- Create, Retrieve, RetrieveMultiple, Update, Delete, SetState, Assign, Route, Merge, DeliverIncoming, DeliverPromote, Send
- **Registering:**
 - `PluginRegistration` tool provides GUI; `PluginDeveloper` tool is for programmatically registration; Source code for both is provided in SDK. (Don't forget to add a Step to the Plug-in.)
- **Deployment:** Deploy to database (recommended) or file system. Debug by providing PDB file in `server\bin\assembly` folder.

IPluginExecutionContext Interface

Properties

- `BusinessUnitId` : Guid
- `CallerOrigin` : CallerOrigin
- `CorrelationId` : Guid
- `CorrelationUpdatedTime` : CrmDateTime
- `Depth` : int
- `InputParameters` : PropertyBag
- `InvocationSource` : int
- `IsExecutingInOfflineMode` : bool
- `MessageName` : string
- `Mode` : int
- `OrganizationId` : Guid
- `OutputParameters` : PropertyBag
- `ParentContext` : `IPluginExecutionContext`
- `PostEntityImages` : PropertyBag
- `PreEntityImages` : PropertyBag
- `PrimaryEntityName` : string
- `SecondaryEntityName` : string
- `SharedVariables` : PropertyBag
- `Stage` : int
- `UserId` : Guid

Methods

- `CreateCrmService()` : object
- `CreateMetadataService()` : object

Technologies

- Good tech skills to have/know:
- CRM SDK
 - C# / VB.NET
 - OOP
 - JScript
 - T-SQL
 - ASP.NET
 - .NET Frmwrk
 - HTML/DOM
 - XML/DOM
 - SSRS
 - IIS Admin
 - Active Dir.
 - Win Security
 - Workflow Fn.
 - SharePoint
 - BizTalk
 - SOAP
 - Exchange
 - CSS
 - SSL
 - NLB
 - MS Oper Mgr
 - Virtualization

Workflow Assemblies

- Pass and retrieve values from custom assemblies in CRM workflow.
- In Visual Studio, create a Workflow Activity project. Reference CRM SDK assemblies and `Microsoft.Crm.Workflow`. Sign your assembly.
- **Basic workflow class example:**

```
[CrmWorkflowActivity("Workflow Step", "Group Name")]
public class Activity1 : SequenceActivity {
    protected override ActivityExecutionStatus Execute(
        ...
        ActivityExecutionContext executionContext {
    IContextService contextService = (IContextService) ...
        executionContext.GetService(typeof(IContextService));
    IWorkflowContext ctx = contextService.context;
    ICrmService crmService = (ICrmService)context.CreateCrmService();
    // Execute custom logic here...
    public static DependencyProperty myStringProperty = ...
        DependencyProperty.Register("myString", typeof(System.String),
        typeof(Activity1));
    [CrmInput("My String")]
    [CrmOutput("My String Output")]
    [CrmDefault("My Default String")]
    public string myString {
        get { return (string)base.GetValue(myStringProperty); }
        set { base.SetValue(myStringProperty, value); }
    }
}
```
- **Deployment:** See plug-in deployment. Restart CRM Async Service.