

MICROSOFT DYNAMICS™ CRM 4.0

SEPARATING DATA AND OPERATIONS WITH MULTI TENANCY



WRITTEN BY:

PETER R. CHASE - EXECUTIVE VICE PRESIDENT, SCRIBE SOFTWARE CORPORATION
PHILLIP HAASE – TECHNICAL EVANGELIST- DYNAMICS CRM, MICROSOFT CORPORATION

MARCH 2008

PUBLISHED BY SCRIBE SOFTWARE CORPORATION 2008 ©. ALL RIGHTS RESERVED.
REPRODUCTION WITHOUT PRIOR WRITTEN PERMISSION IS STRICTLY PROHIBITED.

Microsoft Dynamics CRM 4.0

Multi Tenant Deployment Model

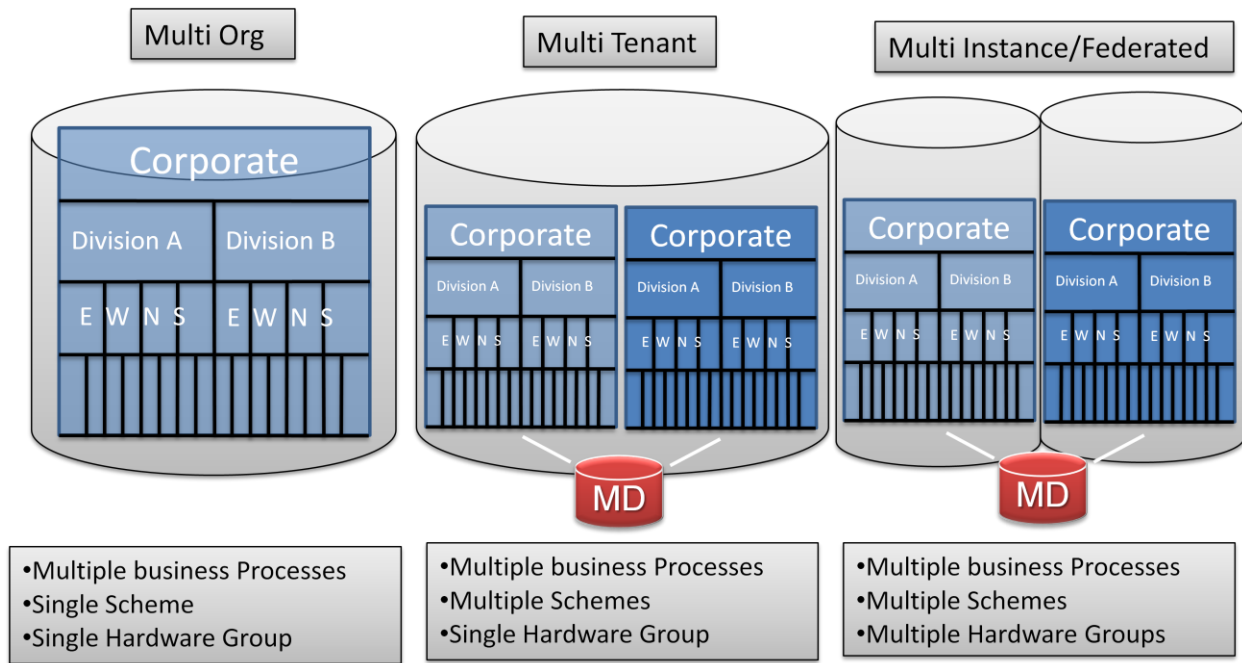
Overview

Multi Tenancy is a new feature of Dynamics CRM and is designed for the separation of data and operations while allowing for a single group of application servers and a single database engine. The primary use is for the hosting of the application where there is a need to separate the data and configuration of an instance from another but still utilize the same hardware group. This results in a model where each tenant is completely separate from one another with regards to data, configuration and process. These multiple tenants can be housed on a single application server and database server or multiple load balanced application servers and single database server. This is a hybrid of separation of data in a single instance and having completely separate instances that do not leverage the same hardware set. Multi tenant allows for the load balancing of all requests from all users to all tenants across a single set of application servers. In addition to load balancing a single failover configuration at the database can serve all tenants. These benefits apply to not only hosting partners but also enterprises where there is a need to separate data and functionality.

Enterprises have business needs that require the separation of process and data. This is most common between geographic or functional divisions. The east coast and west coast should not be allowed to view each other's opportunities. The plastics division has a different sales process than the metals division. These common scenarios create the need for separate data and process within the same application. As you get higher up in the organization the need to see this data rolled up is critical for effectively running the business as a whole. In addition there are times, in the above scenario when the plastics division needs to know if the metals division has activities with the same customer. This need means that there will be common data, like account information, that both groups will be able to see and will be the basis for the roll up reporting to upper management. Dynamics CRM and other line of business applications use a technique known as multi org to separate the data and process. This is done by assigning a person to a position within an organization and then tying data like opportunity, case, account, contact and the like to an organization level. The usual behavior of the application is to allow a person access to information located at or below their level. This technique is well understood and has great benefits when there is a need to implement common practices across business units. With this model it is easier to have common data and practices than have differing data and practices. That being said it is very common to have different processes it just becomes a management task. Generally speaking, the greater the differences the more difficult it is to model in a single instance of the application. This being true it is always less expensive and time consuming to support multi-org than multi tenant or multi

instance/federated. We should only pursue multiple instances when the business needs outweigh the costs. Let's step thru the decision criteria that would facilitate the need for multi tenant or multi instance. The diagram below shows the three models for the separation of data and process.

- Multiple business units, processes or schemes leveraging roll-up reporting
- Leverage single hardware group or separate hardware for geographical needs
- Multi Tenant and Multi Instance/Federated assume master data (MD) sync



Business needs around data

There are times when a business needs to have its schema vary by business unit. This happens when the definition of a data entity is different between units. An example of this would be if a company had two divisions that both need to represent household. The first business unit would need to track which customers are related to an employee to qualify for a company discount. The other business unit needs household to represent a taxable entity for finance purposes. In this case two schemes would be needed and the same objects would need to have completely different representations in both. This would drive toward the need for multi tenant or multi instance/federated. The second area of data is where we have conflicting business rules around the data. An oversimplified example would be the service unit cannot delete an account with an open case and the sales unit cannot delete an account with an open opportunity. In an example like this if we cannot have both units agree to allow a combination of the two rules then we would need to have separate schemes. Another version of this is that we have a call center that is doing warranty repairs for manufacturers. If the different manufacturers designate that the data cannot be mixed or that the schemes of the data need to be different then this would propagate the need for multi tenant or multi instance/federated.

In some scenarios these differing business needs could be combined from an IT or technology point of view but the business demands dictate that they be separate for the case of allowing one unit to be able to move separately from another. This need for agility is weighed against the additional cost to reach a decision. For the examples above the need is present for multi tenant or multi instance/federated. Now we need to assess the additional cost of using more than one instance and if the business and functional value is higher than the cost then we should pursue multi tenant or multi instance/federated.

Business needs around functionality

Functional differences between business units can almost always be solved with the application and extra logic or development around the individual processes so that the logic is executed for a specific business unit. Therefore it will take a higher burden of proof to justify using multi tenant or multi instances/federated over multi org. There is a point where the additional development costs to keep multiple business processes and logic associated with them for different business units will surpass the cost for maintaining multi tenants or multi instances/federated. An additional area where functional requirements of the business weigh heavily is in windows for rolling out incremental functionality. In a company where one business is based around the holiday buying season no changes to the application can be implemented after June. The other business unit has a buying cycle associated with the summer break of school and therefore cannot have application changes after December which leaves no window for the updating of the application. Multi tenant or multi instance/federated would allow the two different business units to have applications that can be modified in independent cycles from each other.

Since the functional requirements are usually a higher burden to prove that a single instance cannot be used data is definitely the first decision point that we use for the use of multi tenant or multi instance/federated. Functionality is addressed only if data does not require the need to move beyond a single instance environment.

Business needs around performance

There are times where the need to move beyond a single instance is dictated by performance. This is most obvious when geographic locations cannot be cost effectively linked to reduce latency and performance of the remote users. An example is that we have a data center in Europe with remote offices in America and Australia. America can have a fast enough connection to let users connect and execute directly against the systems in Europe but Australia cannot. In this case we would need to create an instance in Australia and replicate data to that instance. In this case multi tenant would not be an option as it leverages a single hardware group. The logical option in this case would be multi instance/federated. There are other cases where we would need to separate due to performance and in most all of these cases the solution would be multi instance/federated and not multi tenant.

Integration Considerations in Multi Tenant Configurations

The individual tenant is operated on its own database within the database engine. This allows for true separation of data among the tenants while allowing for fail over at the database level. A single Active Directory can be utilized across the multiple tenants allowing a user to log in once and then enter and exit multiple tenants that they have been granted access to without having to log in multiple times. While the user is within a tenant they do not have access to the data in another tenant. This is done on purpose so that true separation of data can be achieved. While this is indeed the correct design for hosted deployments we also have the need to be able to see certain data from multiple tenants together in context. This is often needed in roll up reporting aggregation data viewing. For this we will need integration to be performed between the tenants. In addition to the integration for the combination of data there is also an implied integration around common data. This is to facilitate the roll up reporting and cross referencing of information commonly known as master data. We need to look at the most common cases and set a reference for this type of interaction. There are a few features of Dynamics CRM that we will leverage for these scenarios like a single active directory for all tenants and a common SQL Server Reporting Server for roll up reporting as well as workflow that can work externally to a tenant.

Roll Up Reporting

This is a very common requirement for an enterprise that would deploy multiple tenants. The desire is to provide visibility into operational performance of metrics that are shared across the different business lines. For this to work there must be common metrics and business data. This could be as simple as having the same regions defined. Then a report would look at all opportunities from each tenant for the regions in a roll up fashion. This report would then have rolling totals so that the metric of pipeline by region would be a cumulative look across the entire organization. This report would not make much sense if the regions were defined in different ways for each tenant. In this way we would need to have synchronized the data to a level that the roll up reporting works. In a more granular scenario we may want a roll up report of the opportunities for a single customer across tenants. In this case we would need to synchronize the customer list across tenants. By doing this the activities logged against a customer would also be able to be correlated between tenants.

In Dynamics CRM we leverage SQL Server Reporting Server for our reporting mechanism. A distinct advantage of using this reporting engine is that it is designed to work across multiple data sources. With this in mind it has no problem looking across the databases of the multiple tenants that are implemented. Dynamics CRM generates filtered views that include the authorization that individual users have to the data. Since these filtered views are the same across the tenants report creation is very simple. The report is made up of the common filtered views in the different tenants with a group by field designated as well as a total column. The user is then defined in each tenant with corresponding view rights so that the report will only produce data at the correct level for the user position. In this case a

regional manager would only have rights to their region in each of the tenants so the report would only select rows that apply to the region that they are allowed to see. There would be no need to have any logic in the report to control the visibility of the data as that is handled by Dynamics CRM and the filtered views.

Consolidated View of Data from Multiple Tenants

For companies that would like to see information from multiple tenants in a single view we will need to build a consolidated view of the data. An example of this scenario is to display all the activities for a given account. The activities from the other tenants would need to be listed on the same screen as the activities from the current tenant. This implies that the activities in the other tenants would need to be associated with the same account record that is synchronized across tenants. This can be accomplished in multiple ways. The first is to leverage the reporting services which can be viewed within the Dynamics CRM user interface to present a list of other activities from multiple other tenants. If only one other tenant is needed in the result then it would be possible to call the other tenant via the web service interface providing the primary key information from the account record and returning the list of activities. The call to the other tenant will be done under the security of the current user which will need to be an active user in the other tenant. By putting the user in both tenants they can be assigned to the same territory and account visibility so that only data that they are authorized to see will be available. This further simplifies the integration as the connection itself does not need to contain the logic for record visibility. Since both tenants operate off of the same Active Directory a user will not have to log on twice to achieve these results.

Master Data Synchronization

In the above two integration scenarios there is an assumption that some data will need to be synchronized between the two tenants. The topic of master data is not new to the enterprise IT arena and it is not new to Dynamics CRM. Although multi tenant is a new feature the same techniques for master data that we have employed in the past will apply to multi tenant. There are two basic techniques to the process of master data. The first is that a copy is never updated, only the master. In this case when a change to master data needs to be initiated by a user in a tenant that is a copy then the change is captured and applied to the master. A separate operation, usually a batch of the changes, will be sent out to the copy. In this case it is common to have a form that captures the changes be the beginning of a workflow. This workflow will allow for the capture of the data and then wrap any processing rules around the change before applying it to the master. In many cases like account creation there are many extra tasks that go with the entry of the data like territory assignment, billing status and categorization of the account within the existing taxonomy. This “extra” data around the account is needed along with possibly an approval process and address verification. After these happen and the date is entered the master needs to do a replication out to the other systems that use this data. The

second technique is to update the copy but mark it as in process. A workflow or integration would then be fired to update the master and go thru some of the same checks and updates as described above. The data would then be pushed back out to the copy to update the existing data and remove the in process flag as well as distributed to other systems that need the data. This process allows the entry of data against the new data before the master is updated. The primary difference between these two techniques is the complexity. As the complexity increases so does the cost and time to maintain the solution. As the number of tenants or other systems increase the cost also rises. The best way to keep the costs down and address the complexity is with an integration technology solution. The category of Enterprise Integration Applications is rich with products that fit the needs of many different customers. Microsoft's BizTalk product is a member of this category and has an adapter that allows the connection of Dynamics CRM to BizTalk. In addition to Microsoft we also have partners that offer solutions to connect Dynamics CRM with other Microsoft Dynamics. Scribe Software is a Microsoft partner that has integration technology to connect Dynamics CRM tenants to other Dynamics CRM tenants as well as other applications. Scribe provides a purpose-built software toolkit (Scribe Insight) to configure, not code, predefined templates. These templates deliver on a company's specific integration processes precisely and quickly. This template configuration model also enables the enhancement and modification of the environment with little or no cost and disruption.

Technical Solution

SCRIBE INSIGHT

Beyond the data, functionality, and performance drivers for utilizing a multi-tenant CRM approach, organizational and political issues may have a significant impact on this decision. In many cases, deployments that involve different groups or divisions in an organization become a matter of negotiation. Questions of who "owns" the data, when data can be shared, who the data can be shared with, and what shared processes will differing groups within the organization participate in can be difficult to navigate in a large enterprise. Answering these questions and delivering a solution that meets the requisite functionality is critical to the support and adoption of the CRM applications by users which is the most important determinant in the success of any CRM deployment.

Given the need to deliver on a set of very specific integration requirements and keep the costs of CRM federation or multi-tenant deployments down, the utilization of custom code or more general integration frameworks are generally not practical approaches. Not only does it take too long and cost too much to deliver on the requirements, but these solutions are hard to change as the needs of the business change or the application environment is upgraded. The ideal approach for delivering a federated or multi-tenant CRM solution is to utilize a purpose-built, template based, software toolkit, such as Scribe Insight, that helps the implementation team deliver on a company's specific integration processes precisely and quickly.

Specific to the deployment methodologies outlined in this document, Scribe Insight provides the following built-in technology capabilities that ensure a successful integrated solution:

Change capture and event automation – The sharing of data and initiation of processes across multiple CRM tenants begins when a change is made to specified data elements within any one of the CRM databases. Dynamics CRM utilizes a capability called plug-ins that will “publish” this data in the form of XML for processing. The capability to configure these call-outs with a few mouse clicks and organize these XML documents for processing across the other CRM tenants is important. Once these changes are published, an automated, fault tolerant, queue-based process can then be automatically initiated to apply the changes to the other CRM tenants. It is also important that the plug-ins are configured to ignore changes that come from another “federated” process to avoid the endless bounce back of changes across the CRM tenants.

Support for varying latency – Different integration processes are going to have very different latency requirements. For example, in the case of synchronizing data in multiple directions between CRM tenants, latency should be as close to zero as possible, especially if the volume of changes to the data is high. This minimizes the chances that changes to the same record will pass by each other in process, creating inconsistent updates across CRM tenants. In cases where data is being replicated in one direction and the time sensitivity of the data is not high (for example, the sharing of past activities across CRM tenants), it may be much more efficient to process these records in large batches during off hours. In any event, Scribe Insight provides the ability to dial the latency of individual integration processes up or down depending on the business need.

Conflict detection and resolution – As mentioned earlier, ownership of data is a very important consideration for any federated CRM deployment. There may be certain data elements that can only be updated by specified CRM tenants. Scribe Insight provides the capability to compare a date and time stamp to ensure that only more recent changes are updated (this can mitigate the issue of changes passing by each other outlined above.) There may be a requirement to only update null values in the target application, so that existing data is not overwritten by another CRM tenant.

Maintaining relational integrity – The relationships, within each CRM tenant, are maintained by a series of unique primary keys for each record. When the same record (an account record for example) is maintained across different CRM tenants, each of the instances of that record will have its own unique primary id. Maintaining a cross reference of these keys for all instances of the record across the CRM tenants is critical to ensure the relational integrity of records within each CRM instance. For example, in the case where a new opportunity is created in one CRM tenant, when that opportunity is processed within another CRM tenant, the foreign key that identifies the account in the source needs to be replaced with foreign key for the account record in the target to ensure relational integrity in the target. Dynamically maintaining the primary id relationships across CRM tenants is essential and is an inherent feature in the Scribe Insight toolkit.

Duplicate detection and resolution – There is not bigger enemy to user adoption than duplicate records. A user will quickly get frustrated by a CRM application if they have to hunt through a significant number of duplicate records to use the system. When new master records are created in one CRM tenant, it is important to be able to ensure that the record does not exist in the other CRM tenants before a new record is inserted. The use of “fuzzy” logic to identify duplicate account and contact records is an important capability here.

Data mapping and transformation – The most obvious need here is the mapping and translation of data elements across the different CRM tenants. Another important requirement is the mapping and cross-referencing of different pick-list values across different CRM tenants. For example, one CRM tenant may have a different set of sales stages than another, requiring a “best fit” mapping between the two tenants. In some cases, multiple CRM tenants can have different database designs requiring some level of structural re-mapping of data. For example, one tenant may have designed a many to many relationship between contacts and accounts, while the other tenant does not. Object-level mapping, as provided by Scribe Insight, will be required to resolve the design differences between the two.

State management – The integration process needs to be able to dynamically update the state of records and transactions within each of the CRM tenants in real-time. For example, a record that was changed in one CRM tenant could have a state value of “updated, not yet synchronized” until all other CRM tenant subscribers to that data had been successfully updated. At that point, the integration process could modify the value of the record to “synchronized”. By having Scribe Insight maintain state at the end points, users are better informed, greater fault tolerance can be designed into the integration processes, and resolution of error conditions is simplified.

Security / record ownership – The integration process should fit within the existing security model of Dynamics CRM, taking advantage of the pre-defined roles, permissions, and data ownership. This ensures data access is controlled and data integrity is maintained. With the ability to initiate data integration processes using the privileged user option, Scribe Insight supports pre-defined security roles/permissions already in existence within the Dynamics CRM environment.

Diagnostics, monitoring, and remediation – Integration of data and business processes across multiple CRM tenants involves dependencies on network and application availability as well as the potential for user actions that were not designed into the integration processes. These are two common scenarios that can lead to exceptions and errors in even the best designed integration processes. Having the capability to proactively monitor for exceptions, anomalies, or inconsistent data conditions and raising alerts to administrators when they occur, as Scribe Insight provides, is essential.

Technical Details

This section explains key details about Dynamics CRM multi tenant deployments.

Operational Setup for Multi Tenant

One of the features of the Dynamics CRM 4.0 is the provisioning of a tenant. This will allow you to have a 'reference' instance provision a new tenant. Each tenant would be identical from a functionality stand point and can include things like identical picklists and seed data but would be separate from a data stand point. Each tenant will need to have configurations applied separately. Because each tenant can be configured independently of each other there will need to be policy put around the deployment of configurations if a single configuration is desired across all tenants. In that case the changes could be made to a reference tenant and then packaged up to be delivered to each additional tenant. The new features in Dynamics CRM 4.0 around extracting a configuration change will make this easy by putting all changes into a single package and not separating out configuration, data and workflow like previous versions.

Data Separation

Each Dynamics CRM organization (tenant) deployed as part of a multi tenant deployment is associated with a unique Microsoft SQL Server database. So a SQL Server database server hosting five Dynamics CRM organizations would have five separate databases, and each database would be configured according to the needs of the associated organization.

Organizations in a multi tenant deployment do share the same web server, but each organization is associated with a unique web application, with separate files and associated settings.

Important: Data and metadata are not shared across Dynamics CRM web applications.

Security

Each Dynamics CRM organization has unique security and access settings, which are stored in the organization's CRM database. User data and preferences are not shared across organizations, and users in one organization cannot access (through the Web UI, CRM APIs or direct database queries) data that belong to another organization.

Note: Active Directory users can belong to more than one organization, but this scenario would require a unique record for that user in each organization's CRM database.

Multi Tenant support in CRM 4.0 also provides for **Deployment Administrators**: users with the privileges necessary to create and manage organizations across the multi tenant deployment. By default, Deployment Administrators are the only users with access to all organizations in that deployment.

Integration

Organizations in a Dynamics CRM multi tenant deployment are not natively integrated. Integration tools that create a consolidated view of the data, especially for reporting purposes, are available from third-party vendors, such as Scribe Software Corporation.

Maintenance

Manage the organizations in a Dynamics CRM multi tenant deployment by using the Deployment Manager application installed on the CRM Server. In CRM 4.0, Deployment Manager provides the following additional functionality:

- Creating a new organization
- Importing an existing organization (from a SQL database)
- Managing Deployment Administrator users.

Scalability

Spreading users across multiple organizations (rather than consolidating them in a single organization) has a slight impact on performance. The overhead results primarily from:

- CRM web servers maintaining information in memory for the different organizations.
- SQL Server overhead of optimizing and executing queries against multiple databases.

CRM MUI Packs

CRM 4.0 provides the ability to configure languages for users based on user preferences. The MUI Pack (containing all the necessary resources) must be installed on the CRM Server on which the organizations are hosted; and then the MUI functionality must be enabled for each organization in a multi tenant deployment. Separating the installation and configuration processes allows organizations to remain completely independent.

Summary

The choice of multi tenant for the enterprise serves up yet another deployment choice following Microsoft Dynamics CRM's philosophy of giving the customer the choice of how best to deploy and utilize software. Now customers can choose how they want to separate their data, configurations and data administration. This can all be done while maximizing the utilization of system hardware and administrative tasks.

About Microsoft

Founded in 1975, Microsoft (Nasdaq “MSFT”) is the worldwide leader in software, services and solutions that help people and businesses realize their full potential.

About Scribe Software Corporation

Scribe Software Corporation provides data integration and migration software solutions to help people within an organization share customer data, to better focus - and take action - on critical business events. By helping everybody within an organization work as one, Scribe helps its customers sell more and provide better customer service. With over 7,500 customers, Scribe is the leading provider of mid-market integration solutions. Scribe Software, founded in 1996, is a privately held corporation headquartered in Bedford, New Hampshire, USA.

Scribe Software Corporation

Three Bedford Farms Drive
Bedford, NH 03110 USA
Tel: 1.603.622.5109
Fax: 1.603.622.3862

Email: info@scribesoftware.com
www.scribesoftware.com

About the authors

Peter R. Chase is Executive Vice President and founder of Scribe Software Corporation. With over two decades of experience in the software industry, in his capacity at Scribe, Peter has advised numerous CRM vendors as they formulated their strategic integration strategies. He has also worked with hundreds of Scribe’s customers to ensure a successful rollout of their enterprise integration solutions.

Phillip Haase is the Technical Evangelist for Dynamics CRM at Microsoft Corporation. Starting at Microsoft in 1990 Phillip has spent the last 9 years in the CRM enterprise industry which included a 4 year position at Siebel focusing on integration. In his current role within the Dynamics US Field organization he serves as a virtual member of the corporate teams working with customers and partners to achieve the optimal business value of technology.