

Oracle® Trading Community Architecture

*Modeling Customer and Prospect Data – TCA
Best Practices V.2*

*An Oracle White Paper
January 2004*

Oracle Trading Community Architecture

Modeling Customer and Prospect Data - TCA Best Practices V.2

EXECUTIVE OVERVIEW

Fragmented, incomplete, inaccurate data has caused headaches within IT organizations for years. As any IT professional can attest, consolidating, cleansing and correcting data for use by transactional systems or for management reporting is time consuming, costly and often virtually impossible to achieve. Oracle acknowledges this issue, and is tackling the problem from a variety of fronts. One area that has been addressed with a great deal of success is enterprise data related to customer and prospect information. In 2000, Oracle introduced a single, centralized, architecture, known as Oracle® Trading Community Architecture, which provides implementing organizations with one global repository for all data related to members of a trading community, whether that data is generated within Oracle 11i, or elsewhere. This architecture is significantly different from previous customer models such as those found in older versions of Oracle applications as well those found in most other vendors' ERP and CRM applications. By leveraging an architecture that maintains a single repository for customer, prospect, partner, competitor, employee, and eventually supplier data in robust and scalable manner, Oracle mitigates the risk associated with fragmented, inconsistent data, and thus provides for more accuracy in transacting and reporting across the implementing organization.

This document focuses on providing structured guidance on the most effective way to model prospect and customer data in the Oracle Trading Community Architecture given the deploying company's unique business requirements and objectives, in an effort to fully exploit the benefits of a global, single customer master repository.

The target audience for this document is primarily external customers who are implementing or upgrading to the 11i E-Business Suite, and consultants who are implementing the 11i E-Business Suite on behalf of external customers. In addition, customers who are leveraging TCA as a Customer Hub solution will also find the content outlined herein beneficial.

For those who are familiar with the first version of Oracle's TCA Best Practices document, you will notice that this revised version holds the same fundamentals as its predecessor, and was built upon the same methodologies and axioms. However, this document has been updated to reflect the latest in TCA and E-Business Suite functionality, and has been revised to denote modifications and enhancements to the TCA Best Practices based on further integration across the E-Business Suite.

INTRODUCTION

The Need

The Oracle 11i Trading Community Architecture is a highly flexible schema that allows you to fully model real world entities in your trading community and accurately represent the complex relationships among those entities. While the flexibility allows the model to better represent reality and enables richer functionality, it also presents the Oracle customer with many alternatives when setting up customer and prospect data in your trading community. These set up decisions are inherently challenging because there is not merely one correct way to model your data; the optimal set up will depend upon the implementing organization's objectives and business practices, whereby a particular customer may be modeled much differently for deploying organization "x" than it is for deploying organization "y". However, the model was designed with certain intentions and concepts in mind. Therefore, while there is not one authoritative answer to how a customer should model their data, there are best practices based on the intentions of the model and the implications of modeling data in different ways.

Please note that the term TCA is often used in reference to two distinct definitions; one definition of TCA relates to the underlying schema, or the "architecture" within the Oracle E-Business Suite that the base functionality is built upon, and the other definition relates to the way in which customers are "modeled" within this schema. Throughout this document, we will refer to the Trading Community Architecture (TCA) when describing the underlying schema and functionality in which a great deal of E-Business Suite applications take advantage of, and will refer to the way in which prospects and customers are modeled within this schema as part of the Trading Community Model.

The Solution

This document addresses the need described above by providing best practices for setting up your trading community, including the relationships that exist with your prospects, customers, partners, competitors, employees, and eventually suppliers. Specifically, the best practices are offered in the form of a process that can be followed to help guide your set up decisions. It is intended to supplement, not replace, your current implementation preparation. The best practices do not address every decision that needs to be made. Rather they focus on the core set up decisions that have the most significant, far reaching implications.

While the best practice recommendations offered in this document are based on the current Trading Community Architecture, they assume the perspective that it is beneficial to choose a set up now that is scalable and that will best position your organization to take advantage of E-Business suite functionality well into the future.

Overview of the Document

Overview of the process recommended in this document

The recommended process entails answering a set of key questions about your prospect and customer data. This document offers guidance on the order in which these questions should be asked and on how to answer each question given your objectives and business processes.

How the document is organized

The document begins by explaining why the Oracle Trading Community Architecture (and its associated model) was designed and identifies the major requirements it satisfies. It then presents the model's core concepts and entities to provide a baseline understanding from which to proceed. While this background knowledge is necessary, the document then goes a step further by distilling the information about Oracle Trading Community Architecture down to the specific information that is most critical for achieving an optimal set up for your data. This information is presented as a list of Trading Community "Axioms" or golden rules. The document then presents the process for determining your optimal set up and the best practice recommendations that are based on the Trading Community axioms. The final section applies the best practices to four sets of example data to show how the recommendations can be used in practice. While the real "meat" of the best practices are within the final two sections, the earlier sections provide the information necessary for understanding and correctly interpreting the best practices presented at the end.

Definitions and Terminology

Confusion often arises merely because people use terminology differently. For example, some companies use the term "customer" to represent both current customers and prospects that might later become customers. Other companies use the term "customer" to refer only to people who have already purchased from them.

For clarity, the following terminology is used in this document:

- "Deploying company" refers to the Oracle customer that has, or will, install, implement, and run all or part of Oracle E-Business Suite. The audience for this paper includes employees of the "deploying company" who are implementing the applications for their company or consultants who are implementing the E-Business Suite on behalf of the "deploying company".
- "You" refers to the person trying to model the data. This might be the external customer deploying Oracle E-Business Suite or a consultant setting up data on behalf of the deploying company.
- "Customer" is used to represent a person or an organization with whom the deploying company has a selling relationship, regardless of whether anything has actually been purchased or serviced. Note, a selling relationship may be

established by actually selling products or it may be established simply by negotiating terms that will be used if you later sell products. In both scenarios, a "selling relationship" exists.

- “Prospect” refers to a person or organization with which the deploying company does not yet have a selling relationship. A “prospect” may or may not become a “customer” at a later point in time.
- “Party” is an entity in the Trading Community Model that can enter into business relationships. A party is a real person, organization, branch, subsidiary, legal entity, holding company, etc. The attributes of a party are universal. In other words, they are independent of your selling (or ultimately buying) relationship with the party.
- “Account” refers to the details of the deploying company’s *selling* relationship with a particular customer.
- A “Hierarchy” refers to a group of parties that have been linked together via party relationships, defined as “hierarchical relationships”. An example of a hierarchy includes a “corporate hierarchies”, whereby an organization’s headquarters is modeled as a party at the top level of the hierarchy, with all divisions and branches linked via hierarchical relationship types throughout the organizational structure. This model facilitates the ability to take advantage of “rollup” functionality, which looks at the related entities in a hierarchy, and “rolls up” the requested data linked to each party. *Note that although some applications have begun to expose this functionality, not all applications have incorporated this as of yet. However, by taking advantage of TCA Hierarchy functionality, you will be well positioned to take advantage of extended application functionality as it is provided in the future.*
- In an organization chart, such as the one below, each box on the chart will be referred to as a “business entity”, regardless of whether it is actually an HQ, branch, division, etc. Each business entity will be linked hierarchically or laterally via “relationships” (e.g. division of, branch of, etc.). The term “company” will be used to refer to the entire business including all business entities in the organizational structure.

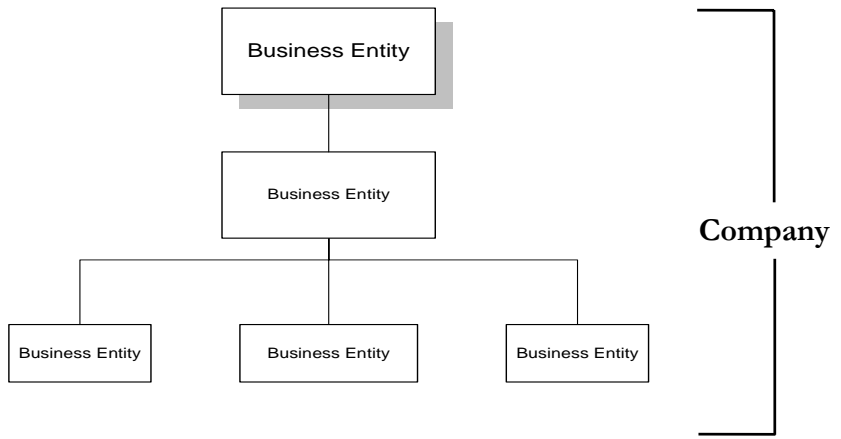


Figure 1: Terminology to describe a business organization

THE TRADING COMMUNITY MODEL – BACKGROUND

Why the Oracle Trading Community Architecture

Today's economy requires business strategies and relationships that are both complex and dynamic. The static and independent customer and supplier models typical of other ERP and CRM applications cover only a limited slice of the trading community and cannot adequately represent the reality of today's complex business communities. The Oracle Trading Community was designed to address these gaps and provide a comprehensive, flexible architecture that can accommodate all players in a trading community and the complicated relationships among them.

Moving from customer and supplier models to a trading community

In the past, different players in a trading community have typically been modeled separately. For example, if company xyz is both your customer and your supplier, most ERP and CRM applications would store two separate representations of company xyz with no link between them. Furthermore, most ERP and CRM applications would store these entities in completely separate places in the database, thus proving it very difficult to know where two entities may be actually be the same. This fragmented architecture prevents you from knowing, and thus benefiting from, the complete relationship you have with your trading partners. Oracle moved beyond these independent models by creating Oracle Trading Community, an architecture that supports linking any and all types of relationships to the same single representation of an entity enabling a full, 360-degree view of your trading partners. By redesigning the Oracle customer model into this new, expansive architecture, Oracle has laid the foundation and has taken a major step towards its vision of an entire, integrated trading community.

Supporting people as well as organizations

Customer models typically cater to the Business-to-Business paradigm or the Business-to-Consumer paradigm, but not both. Business-to-Business customer models often lack support for individual consumers while Business-to-Consumer customer models often cannot accommodate organizations. The Oracle Trading Community Architecture recognizes the differences between individuals and organizations and includes distinct, comprehensive support for both. Depending upon whether you want to do business with an individual or an organization, the architecture allows you to capture the relevant information. For a firm, this might include its legal structure and financial data. For an individual, it could include personal interests and employment history. The ability to fully model both individuals and organizations makes TCA an appropriate solution for both Business-to-Business and Business-to-Consumer paradigms. In addition, Oracle facilitates the modeling of HR employees in the Trading Community Architecture, allowing for instances where employees of the deploying organization are customers, or related to customers of that company. This piece of functionality

truly enhances the 360-degree view in depicting an integrated model of an organization's trading community.

Modeling complex relationships

Especially with the extended supply chains common today, it is not enough for companies to know only about their own trading relationships. Competitive advantage often requires identifying and understanding relationships between other members of the business community – relationships that the deploying company is not a part of. For example, the deploying company might want to identify its customers' customers or even its partners' customers. Or, the deploying company might benefit from knowing which other companies are customers of its current supplier for a particular part or service. Oracle recognizes the importance of these types of relationships and provides support for them in the Oracle Trading Community Architecture. Oracle also recognizes that real world relationships are rarely simple and thus extends its support to include the modeling of complex relationships such as corporate hierarchies and exchange scenarios.

Providing one integrated solution for the entire E-Business Suite

Oracle's Trading Community Architecture includes the single customer model that supports the entire E-Business Suite and is key to the integration that makes Oracle Applications so powerful. The Trading Community Architecture enables the seamless, coordinated processing of key business flows such as "Campaign to Cash" or "Problem to Resolution" across the complete E-Business Suite. While all Oracle Applications share the same model, each exposes different parts of the model to different degrees depending upon the context and the need for particular information.

Anticipating Future Directions

While some E-Business Suite Applications have not yet fully exposed the potential offered by the Trading Community Architecture, most now do. In fact, since the first release of this paper in 2001, the number of E-Business Suite applications that expose elements of the Trading Community Model has increased significantly and the functionality provided by the Trading Community Architecture has grown exponentially. However, if you are still waiting for Oracle to provide a particular feature that would harness the power of the Trading Community Architecture, or if you simply have an affinity for the "way it was before," you may be tempted to model your data "the old way" with the R11 data model in mind. Understanding that functional needs of today often drive implementation decisions, the importance of positioning yourself for the future cannot be underestimated. As different applications develop new features, they will, by definition, be doing so in the context of the Trading Community Architecture because the model facilitated by this architecture is the single model for the entire E-Business Suite. Understanding this eventuality and trying to set up your data with the intentions

and axioms of the new model in mind will better position you to take advantage of the new features and functionality in future E-Business Suite releases.

TRADING COMMUNITY MODEL - THE NUTS AND BOLTS

This section defines the core entities in the Oracle Trading Community model and highlights the particular concepts that are key to understanding how to set up your data. It assumes a working knowledge of TCA and thus is not a comprehensive description of the model.

Core Concepts and Intentions

Separation of the entity from your relationship with that entity

As explained above, Oracle's Trading Community Architecture will support linking any and all types of relationships to a single representation of an entity. Entities may include prospects, customers, partners, employees, and in the future, will include other members of the trading community, such as suppliers. This model enables a 360-degree view of the members in your trading community.

Architecturally, TCA makes this comprehensive view possible by separating the entity you are doing business with from your relationship with that entity. For example, if you sell products to Business World, the attributes inherent to Business World would be stored as one entity (a party) and the attributes of your selling relationship with Business World would be stored as a separate entity (an account for the party). Separating the entity from your relationship with the entity makes it possible to point out other relationships, such as the supplier relationship, to that same entity in future releases of the E-Business Suite. It also allows you to identify prospects before you are ready to represent them as your customers, whereby party information will be captured for the organization or person without an account relationship (the account would denote a selling relationship). Understanding this separation of the entity from your relationship with the entity is critical for determining how to set up your data.

Source of Truth Registry

A powerful by-product of the separation described above is that you end up with a "source of truth" registry of information. This registry contains universal facts about entities, facts that are completely independent from you and your relationship with that entity. This allows you to model external organizations more objectively and provides an appropriate infrastructure for loading externally supplied data. For example, to build your customer base you could purchase a list of new companies from a marketing list provider. By populating the Trading Community Architecture with this data, you create a registry of new "prospects", which you can pursue.

Consistency with Dun & Bradstreet

The Trading Community Model was designed to complement the modeling approach of one of the most widely accepted third-party content providers for business data, Dun & Bradstreet ("D&B"). D&B assigns a unique identifier, the D-U-N-S Number, to an organization based on the organization's name, location,

and line of business. Because this D-U-N-S number is an attribute in the Oracle Trading Community Architecture, you can easily model D&B entities throughout many applications in the E-Business Suite. Furthermore, D&B takes the customer and prospect modeling a step further, by supporting the Party level relationships housed in Oracle, that exist amongst organizations (e.g. global ultimate of, domestic ultimate of, branch of, etc.), thereby showing the linkage of the organizational hierarchies or “family trees.”

Modeling relationships

The Oracle Trading Community Architecture was designed to support not only your relationships but also relationships of which you are not a part. For example, the deploying company might have relationships with two subsidiaries that are both related to a third holding company. While you don't have a direct relationship with the holding company, you might want to know that the two subsidiaries are related to the same holding company. Or, in addition to tracking your own customers, you may want to identify the end users of your product by tracking the relationship between your direct customers and their customers. In TCA you can identify and exploit these types of relationships between other members of the trading community.

Modeling multiple customer relationships for one party

It is very common for a deploying company to have more than one selling relationship with the same customer. For example, different divisions of the deploying company might negotiate their own relationships with the same external party, thus representing these different selling relationships by different accounts. Or, the deploying company might create multiple relationships with the same party if the customer negotiates unique terms to purchase different products or services from that organization. In either scenario, Oracle's Trading Community Architecture allows you to thoroughly and accurately represent these more complex situations by recording multiple accounts with the same trading partner.

Intended Perspective

Every set up decision carries ramifications, but some are more significant than others. In addition, the same implication may be considered a benefit from one perspective and a liability from another perspective. For example, a CRM-only perspective might lead to different set up decisions than would an ERP-only perspective. Because TCA supports all Oracle Applications, it was designed from the perspective of the complete E-Business Suite. Therefore, set up decisions approached with an E-Business Suite perspective will best match the intentions of the model and will ultimately result in the most advantageous data structure.

Logical Model and Entity Descriptions

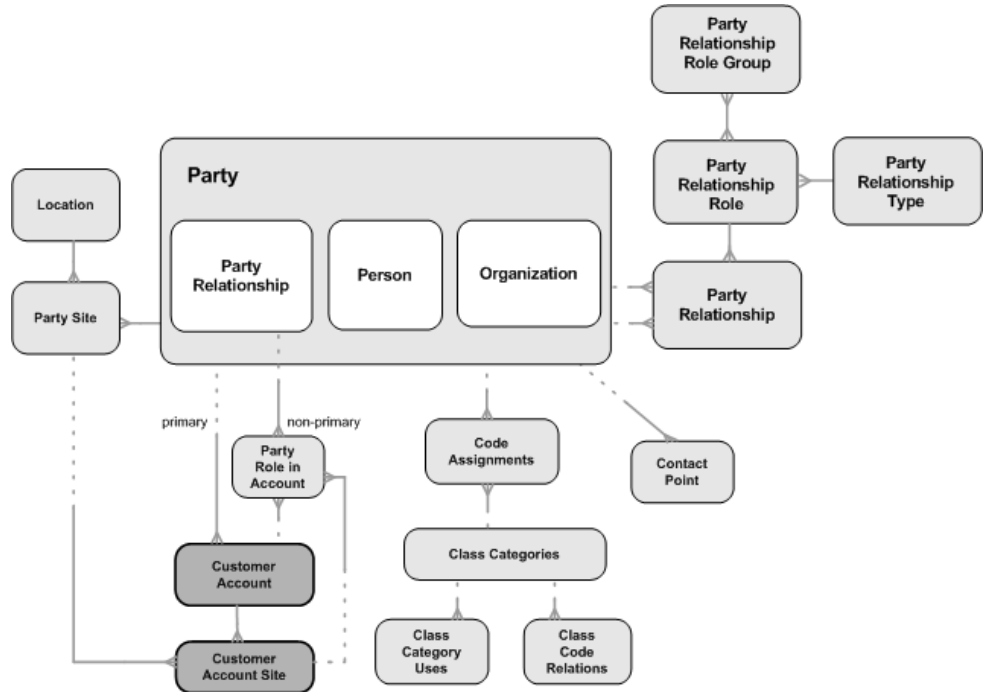


Figure 2: Logical representation of the core portion of the model

Entity Definitions

- Party – an entity that can enter into business relationships. A party is a real person, organization, branch, subsidiary, legal entity, holding company, etc. Any real thing that you would want to put a name to is a party. The attributes of a party are universal. In other words, they are independent of your selling (or ultimately buying) relationship with the party.

There are different types of parties such as:

- Organization
- Person
- Party Relationship
- Party Relationship – A binary relationship between two parties such as a contact, employee, etc. Party relationship types can be seeded or user-defined. A party relationship is optionally a party itself, meaning certain party relationships can engage in various transactions across the E-Business Suite. The following example is indicative of Party Relationship functionality: Joe is an individual consumer of Business World, purchasing B2C goods directly for his personal use. Business World stores information on Joe such as his home address (for billing and shipping) as well as his personal email address and

phone number. XYZ is a B2B customer of Business World, and purchases goods on behalf of their entire organization. Business World stores information on XYZ such as their global purchasing agreement, their bill-to and ship-to information, and additional organizational information (e.g. website, phone number, etc.) Joe works for XYZ, and is linked via a Party Relationship. The information Business World stores on Joe at XYZ is separate from the information stored on Joe as a Person or the information stored on XYZ as an Organization. As a Party Relationship type “Employee OF”, Business World stores information such as Joe’s position at XYZ, his contact information (e.g. phone number, email, etc.) at XYZ, as well as his address at XYZ. Joe, as an Employee of XYZ can enter into transactions with Business World separately from Joe as an individual. For example, if Joe calls into to place a B2C order for his own personal use, Business World would record this transaction at the “Joe” level, whereby if Joe called in on behalf of XYZ to place a B2B order, Business World would track this information at the “Joe at XYZ” level.

- Location – a point in geographical space described by a street address.
- Party Site – the link between a party and a location that indicates the location is valid for that party. Party sites should not be used to model the hierarchy or organizational structure of a company. The organizational hierarchy should be modeled using party relationships.
- Contact Point – a means of contacting a party other than sending mail; e.g., a phone number, e-mail address, URL, or fax number.
- Account – the attributes of the selling relationship between the company deploying Oracle Applications and a party. Account attributes do not describe a party; they only exist when a selling relationship is present between the deploying company and the party.
- Account Site – a party site that is used within the context of an account; e.g., for billing or shipping purposes.
- Party Role in Account - allows multiple parties to play various roles in an account; e.g., a “ship to” contact for an account.
- Classifications - a means of categorizing a party into a pre-defined classification scheme. You can classify a party using seeded class categories (such as SIC codes or NACE codes) or you can define your own classifications scheme that is more appropriate for your organization.

To set up data correctly, it is critical that you understand the intended meanings “party”, “party site“, and “account”.

As defined above, a “party” is meant to represent a real thing with which you can do business and a “party site” is simply meant to identify that a particular location is valid for the party. A party site is not intended to represent a distinct business entity with which you can do business.

Note: this is a significant departure from the modeling concepts commonly used in releases of the Oracle Customer Model prior to 11i when customers were modeled within Oracle Receivables. Based on previous experience, you might be inclined to use party sites to represent distinct business entities that you consider “sub-parties” of the actual party. In the customer model prior to TCA (e.g. “AR” Customer model and 3i Sales and Marketing model), as well as in other 3rd party applications, addresses were often used to represent various branches or divisions of the header level customer. However, this usage is not consistent with the definition and intended usage of party sites in the new Oracle Trading Community Architecture.

In addition, sometimes people want to model a company or business entity as an account. However, a business entity should not be modeled as an account because an account is simply the attributes that describe your financial relationship with a party. It may help to think of the following example:

Suppose you purchase a list of organizations from a third-party content provider and the list includes Business World, a company you have never heard of before. In addition to the company name, the list provides data about Business World such as the number of employees, the CEO’s name, and the year established. Because this data is factual information about Business World, it would be the same no matter who purchased the data from the content provider. Thus Business World would be stored as a party, in the “source of truth” registry.

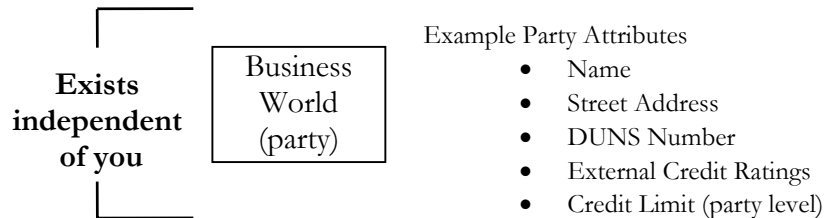


Figure 3: Representation of a party

Since you do not yet have a relationship with Business World, an account is not necessary. However, let’s say three years later you finally establish a selling relationship with Business World. The selling relationship may be established by actually selling products to Business World or simply by negotiating terms that will be used if you later sell them products. In both scenarios, a “selling relationship” exists. To record the attributes of the selling relationship, such as shipping and billing preferences, you create an account for the Business World party.

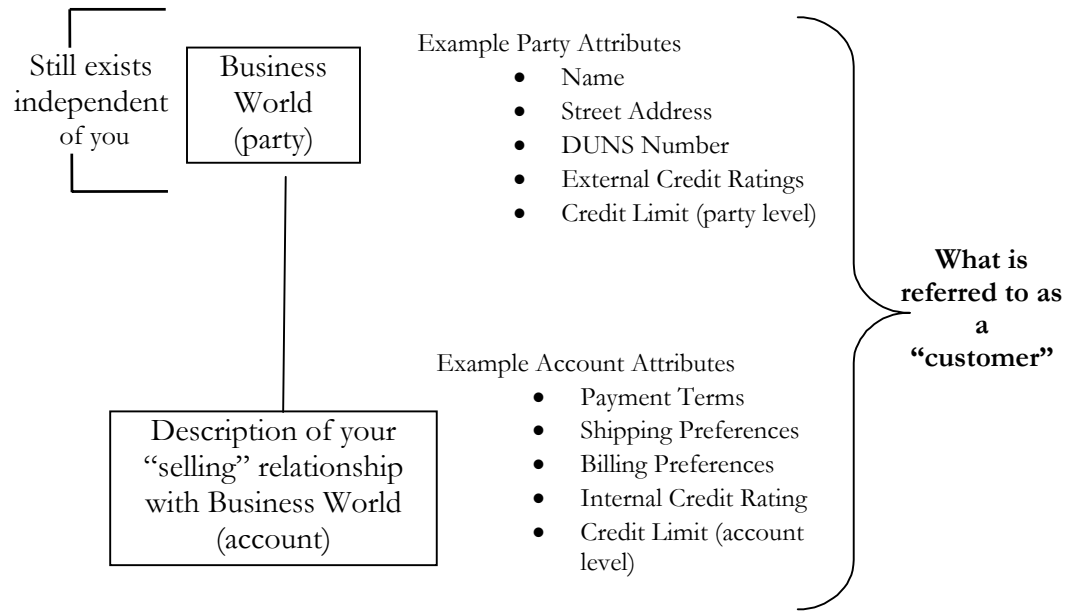


Figure 4: Representation of a "customer"

The Business World party still exists independent of you. However, now that you have a selling relationship with Business World, you record that relationship as an account for the Business World party. The combination of the party and the account represents what is referred to as your "customer".

What you see in different Oracle Applications

Another way to solidify your understanding of these entities is to think about what you might see in various applications. As mentioned previously, different applications may show different subsets of the data from TCA, from different perspectives depending upon the context of their functionality.

In the case where you are using Oracle's full E-Business Suite, a party is usually created first in CRM and an account is created later (in either ERP or CRM) when it becomes necessary. In many cases, the account is automatically created when a party wants to place an order. However, there can be variations to this chain of events. It may be helpful to explicitly identify what is shown in a few applications in an example business flow. The following is what you might see as part of the Campaign to Cash flow:

- In Oracle Marketing Online, the user can create and execute marketing campaigns and events for prospects. In this scenario, prospects are modeled in TCA as parties. Since the deploying company does not yet have *selling* relationships with the prospects, accounts are not needed. The Oracle Marketing Online (OMO) user interface exposes party information as the primary source of prospect and customer information. Via integrated

functionality within OMO, Account information can be accessed; however, this information is not often applicable from a Marketing perspective.

- From a sales perspective, the user can query up these same prospect parties in Oracle Sales Online (OSO). As in Oracle Marketing Online, the user interface exposes party information as the primary source of information on organizations and persons. Alternatively, the user might use Oracle Telesales (OTS) to create leads and opportunities for these prospects. Again the user interface would be exposing party information because leads and opportunities are associated with parties, not accounts. However, you may want or need to deal with Account information when utilizing either of these applications. Consequently, both applications do allow you to create accounts or simply view existing account information.
- If a sales opportunity is successful and a quote is needed, the quote can be created in Quoting. To save a quote, an account no longer needs to exist. As of 11.5.9, Quotes can be created at the Party level without requiring Account information.
- Whether or not Quoting is used, an account is always required to create an order. To place an order in the Sales Order form in Order Management, an account must already exist. You can create this account manually in Order Management or in Oracle Receivable first and then place the order in the Sales Order form.
- Once Order Management has interfaced the orders to Oracle Receivables, the user can view invoices for these orders in Oracle Receivables by querying up both the party and its associated account in the Customer Form and then navigating to the transactions workbench. In the Customer Form you can also create a new account for an existing party or you can enter a brand new party and its associated account(s). The Customer Form displays both party and account attributes to represent “what we think of as a customer”. Attributes of the person or organization that are independent of you, such as the company name would be party attributes. Attributes of your relationship with the person or organization, such as shipping or billing preferences, would be account attributes. It is the combination of the real world entity and your selling relationship with that entity that makes it your “customer” (see figure 4).

THE TRADING COMMUNITY AXIOMS

The previous section of the document provided background information necessary for understanding the Trading Community Architecture, and for determining the optimal setup for your trading community in accordance with the Oracle Trading Community Model. This section offers a transition from the background knowledge to the best practice recommendations made later in the document by distilling the information about Oracle Trading Community Architecture down to the specific “gems” that most directly drive the best practices.

The following axioms are the “golden rules” of Oracle’s Trading Community Architecture. They are agreed upon truths that reflect the objectives and intentions of the model and can be used to guide your decisions in the set up process. While the axioms are just brief statements, they capture the essence of the new model and serve as the foundation for the best practices.

1. A party is a “real” person, organization, branch, subsidiary, legal entity, holding company, etc. Any real thing that you would want to put a name to is a party. The attributes of a party are universal. In other words, they are independent of your selling (or ultimately buying) relationship with the party.
2. Party sites are simply meant to represent the valid set of locations for a party. In technical terms, a party site is merely the resolution of a many to many relationship between parties and locations.
3. Party relationships should be used to construct the organizational hierarchy/structure of a business.
4. An account represents the attributes of the deploying company’s selling relationship with a party. A real thing, such as a person or organization, cannot be an account. Rather, a real thing can be a party and the deploying company’s specific relationship with that party can be represented by an account.
5. An account should not need an account name for the purposes of allowing the deploying organization to identify the organization or person. The name of the entity will be stored at the Party level. The assumption behind this axiom is that names are typically appropriate for identifying real things and an account represents a relationship with a real thing, not a real thing itself. Account descriptions, rather, should be used to identify the different selling relationships that the deploying organization has with a party. Note that at this time, some applications may refer to the Account Description as Account Name. However, the use of this field should remain consistent with this axiom.

Business-to-Business Example (a):

If the deploying company markets and sells to Business World as one global entity, but Business World has two distinct buying relationships with the deploying company; GSA purchasing, and non-GSA purchasing, then two accounts should be created and named accordingly in the Account description (e.g. name) field. This type of denotation at the account level will allow order entry and finance personnel to properly identify which account they should work with for Business World, given the transaction at hand.

Business-to-Business Example (b):

If the deploying company markets and sells to Business World as one global entity, but Business World has purchasing operations in two countries (US and France), one Party may be created to represent Business World, whereby two accounts would be created to represent the two distinct purchasing operations of Business World. In this scenario, different account terms would be represented in each respective account, and the account name for each would denote whether the account is for the US or the French operation. Again, this type of denotation at the account level will allow order entry and finance personnel to properly identify which account they should work with for Business World, given the transaction at hand. (Note that if the deploying company markets and sells to the US and French operations of Business World *separately*, each should be represented as a separate Party with its own account).

Business-to-Consumer Example:

If a deploying bank markets to Jane Doe as one customer, but offers her multiple types of accounts with their bank, the bank may want to refer to one of Jane's accounts as "Checking" and the other as "Savings" in the Account Description field. This way, all Sales and Marketing information related to Jane is stored under one Party, but given that Jane has two accounts, the bank's call center will be able to properly identify which account to work with when Jane calls in (without being provided the account number). While the bank's call center might want to see these "names" internally for these types of scenarios, the bank would not use them to identify the accounts from a transaction perspective. Rather, they would use the account number to denote any activity against either account.

6. An Account represents the billing relationship an entity has with the implementing organization, whereby the bill-to address is just one component of all of the account information that is stored. An account should typically have only one active bill-to site, given that most times, different terms and conditions apply to different bill-to sites. In addition, most often different sales, service, and marketing efforts apply to the party sites that relate to the bill-to account sites. Although from a functional perspective, different terms and conditions can be applied to account sites within certain E-Business Suite applications, the axiom indicates that if you need an additional bill-to site, you typically need another account. You may even need a new Party and Account. For example, from a CRM perspective, you may want to capture unique

interactions with a party site that corresponds to a particular bill-to account site. This would require the creation of a new party and associated account to track this unique relationship, given that account sites are linked to party sites. The rules of this axiom are based on the use of Parties, Party Sites, Accounts, and Account Sites within cross-functional business flows across the E-Business Suite. However, if your organization encounters a situation where multiple bill-to addresses fall under the exact same terms and conditions of another bill-to address for a customer, you may indeed create multiple bill-to sites under one account and still be compliant with TCA Best Practices. For example, if the customer you are dealing with has three different locations or lockboxes, which all fall under the same billing and account parameters, you may choose to create one account with three different bill-to sites. This setup complies with TCA Best Practices because in this situation, you are not intending to transact with or track information separately for each bill-to site. Rather, you are actually intending to use each account site (e.g. lockbox) interchangeably.

Although plausible, the scenario noted above is most often an exception rather than the rule. When faced with the decision as to whether multiple active bill-to sites are applicable for an account, TCA recommends that you strongly consider the implications from a cross-functional business perspective. Please note the following as just one example of the considerations necessary when creating multiple bill-to sites per account.

Almost all CRM transactions are striped according to either the Party or the Party/Account combination. As such, if the deploying company were to implement iStore (or various other E-Business Suite applications) with a TCA model that reflects one Party with one Account and multiple bill-to sites, every time a transaction is placed against that account from the end customer, all other bill-to information will be exposed. As such, any user who has access to the iStore can book a transaction against any of the bill-to sites that are present under the Party/Account combination, or take advantage of any discount associated with the other bill-to sites. From a data integrity and security perspective, therefore, it is recommended that accounts typically have one active bill-to site. Furthermore, from a transactional and reporting perspective, the account is used to capture data from many E-Business Suite applications and thus, provides a great deal of information for reporting. Therefore, by creating accounts for each bill-to site, you are expanding your ability to capture and report off of critical information.

HOW TO SET UP CUSTOMER AND PROSPECT DATA

Recommended Process

If you are familiar with previous releases of Oracle's customer model (previously part of Oracle Receivables), try to let go of these concepts and definitions and, instead, think in terms of the new model. To take full advantage of current and future functionality, you need to set up your data based on the new Trading Community Model concepts and intentions.

Step 1: Answer the key questions about your customer or prospect data in the order the questions are presented. To answer the questions, use the provided recommendations and your understanding of the Trading Community Architecture axioms.

Step 2: Assess the major implications of the decisions made in step 1 to ensure that your decisions are consistent with your objectives. (Note: Implications are identified below in this section of the document).

Step 3: Consider whether your chosen set up will position you well for the future.

Step 4: Go back to step 1 and review the questions again to cover any interdependencies and ensure you are getting the full picture.

The Key Questions and How to Answer Them

The following questions about your prospect and customer data will help you determine the optimal way to model your data. While the list of questions is not exhaustive, it addresses the most critical decisions that need to be made. The responses provided are not specific answers but rather guidance on "how to go about" answering the questions. Use of these questions and responses will be demonstrated with sample data in a later section of the document.

Question 1

Consider each member of your customer model (i.e. each prospect or customer) individually. When considering a particular company (including the multiple business entities that make up its organizational hierarchy), which entities should be modeled as parties? Which should be modeled as party sites?

How to Answer Question 1

Recommendation 1: Create parties for any business entity that you interact with, or intend to interact with as its own entity. For example, if you interact with the Telecom division of Business World as a separate entity from Business World HQ (e.g. different target marketing, different sales representative, etc.), you should model Business World Telecom as its own party. You can link those parties into the company's organizational hierarchy using party relationships.

To determine the level at which you interact with the company consider the following:

- If you will need to distinguish your relationship with one business entity from your relationship with another business entity, the business entities should be separate parties. Another way to think of this is if you are going to need an account specifically for a particular business entity, that business entity should be represented as a party.
- In general, if you want to see activity for one business entity separate from activity for another business entity, the two business entities should be separate parties.
- If you want to distinguish between business entities that you do business with for reporting purposes, the two business entities should be separate parties.
- If you want to categorize a business entity using the classification model within Oracle Trading Community Architecture, the business entity should be its own party.
- If you are modeling a large, global company that has 2,000 business entities should you create one party, 2,000 parties, or some number in between? It depends on how you interact with that company. If you truly interact with the company as one global business entity, then you only need one party. On the other hand, if you want to uniquely identifying and interact with all 2,000 business entities, then you should create 2,000 parties. However, these extremes rarely reflect reality. Most likely you will not interact with the company as one single global entity but you will also not have specific business reasons for interacting with all 2,000 business entities. The answer is generally somewhere in between. To determine the number of parties needed, identify all the business entities that you want to interact with distinctly.

Recommendation 2: Do not represent different business entities as a bunch of party sites for a single party with the intention of using the party/party site structure to represent the organizational hierarchy of a company.

Note: We recognize that you might be inclined to automatically model a company as one party with multiple party sites to represent different business entities that you consider to be “sub-parties” of the company, because that approach was often used in implementations for previous releases. However, that approach is not consistent with this recommendation or with the intentions of the new model and will not position you well for the future. And as described below, the decision of what to model as a party and what to model as a party site has important ramifications.

- According to Axiom #2, “party sites are simply meant to represent the valid set of locations for a party”. Without representing a true organization structure within one party, there are circumstances where many party sites are valid for one particular entity. For example,

suppose you are modeling XYZ organization, which has two divisions, XYZ 1 and XYZ 2. Although each of these entities should be modeled as distinct parties, each party has multiple party sites. In this example, the parent company, XYZ HQ has a primary, bill-to site as well as an alternate bill-to site to represent a different bill to address used for one of its other accounts. In addition, when Business World sales representatives go to XYZ HQ, they must obtain a badge at a security building (represented by a party site), and give product demonstrations at the XYZ HQ Conference Center, (again, represented by a separate party site). Note that in this example, XYZ HQ has four distinct party sites, but none of which represent a hierarchical structure. All sites related to XYZ HQ are valid locations for that party and are used by Business World employees to better know their customer (in this case, XYZ HQ).

- Party sites are not intended to provide a hierarchy for processing. As mentioned in the previous recommendation, the organizational hierarchy of a company should be constructed with party relationships. To provide maximum flexibility, the current party relationship model does allow the modeling of hierarchies in which a party has more than one parent. While this flexibility enables you to accurately model more complex relationships, it does have some implications regarding Relationship Manager functionality. In the latest release of Relationship Manager, only strict hierarchies (e.g. hierarchies that maintain only one parent for each entity) can be seen in the hierarchical view functionality. As such, it is only these strict hierarchies that will work properly for rollup purposes across the various E-Business Suite Applications. Therefore, as parties require multiple parents, the functionality is available to do so. However, please consider the implications outlined herein.

Note: This recommendation suggests that party sites not be used to represent an organizational hierarchy. On its own, and for most part, this statement is a great premise from which to work. However, your organization may not think of a customer in terms of having multiple physical locations that should be modeled by parties, because from a sales and marketing perspective, all you care about is them as a logical entity (e.g. HQ). You may want to know the addresses of their other physical locations, however, simply because your sales representatives visit other physical locations for training seminars, etc. An example of this scenario is as follows: Your business may work with Customer X, which is a division of a large global organization. This division has ten buildings within a complex, each with its own distinct postal address. However, given that you work with Customer X as one entity (not ten distinct entities), you should create one party for Customer X with ten party sites to denote its various buildings. Keep in mind that the way you choose to model Customer X may be completely different than another deploying company, whereby that other company may actually sell or market to all of the sites

for Customer X separately, thus warranting different parties for each. Although this scenario seems to be in line with the old customer model, it is a plausible scenario, and could fall in line with TCA Best Practices. In most cases, it is not an entire company that is modeled by party sites, but rather a particular division or group where this may come about. As such, a corporate hierarchy (“family tree”) will still usually exist as part of the broader hierarchy for this customer, whereby only a particular division (party) is modeled with various party sites if necessary.

Although the model just described does fall within TCA Best Practices, we strongly recommend that before modeling a customer as such, you thoroughly analyze your specific scenario to deem its appropriateness.

Recommendation 3: If you want to map Dun & Bradstreet (D&B) data to your prospect and customer data, entities from D&B should be implemented as parties within Oracle Trading Community Architecture because D&B’s unique identifier, the D-U-N-S Number, is an attribute of the party. You can then link the parties into the company’s organizational structure using party relationships.

Implications of different answers to Question 1:

Whether you model a business entity as its own party or as a party site has significant implications based on the functionality associated with parties and party sites.

Creating parties for the individual business entities allows you to interact at the business entity level. In addition to giving you visibility to your activity with each business entity, it allows you to fine-tune those activities, such as marketing campaigns or sales strategies, to cater to the unique characteristics and requirements of each individual business entity. When you model your prospects and customers at this more granular level (i.e. multiple parties for different business entities in a company) rather than as one big whole (i.e. a single party), you are given the opportunity to know them better, and thus serve and sell to them better.

Here is a sampling of current functionality that is available to business entities modeled as their own parties but that is not available for business entities modeled as party sites.

- **Marketing Activities:** Marketing functions such as events, campaigns, collateral, and market segmenting, are all associated with parties not party sites.
- **Sales Assignments:** Leads and opportunities must be associated to a party but can be optionally associated with a party site for that party as well. When modeling your customers in accordance with the Oracle Trading Community Model, the assignment of party sites to a lead or opportunity is not necessary. The reason is because leads and opportunities are striped to the party, and therefore if your customers are modeled according to the Best Practices, all leads and opportunities attached to a party related to the primary bill-to site (as well as all other relates sites). Although a party site is not required (or even necessarily recommended) for lead and opportunity creation, you will notice

that party sites are displayed for leads and opportunities. The reason for this was to accommodate Oracle customers who were modeled according to the traditional “AR” model, and those using the old 3i Sales and Marketing application prior to the new E-Business Suite Applications. In addition, for customers who have migrated to Oracle Applications from 3rd party systems, where the lead/opportunity address is a key requirement, this field provides the ability for customers to migrate accordingly. However, because sales teams are associated at the party level not the party site level in Oracle 11i, you can’t have different sales teams working on leads and opportunities for different business entities if the business entities are set up as party sites rather than parties. In addition, as you are using this document as the baseline for your organization’s customer modeling standard, this field will likely not be required.

- **Accounts:** An account is created for a party not for a party site. The selling relationship is not with a location but with the party that is using that location.
- **Quotes:** Quotes are created for parties not for party sites, but can be tied to a party site as well. However, when querying for all quotes for a particular organization, this information is attached to the party, not the party site.
- **Contracts:** Service contracts, corporate contracts, and contracts for intellectual property are all created for a party (and its associated account); they can not be created just for a party site.
- **Install Base:** A business entity modeled as a party, with an account, can be the owner, lessor, lessee, service provider, re-marketer, customer, or regulatory agency for an item instance in an install base but a business entity modeled as a party site cannot. A business entity modeled as a party site can only be used as a location where an item instance resides.
- **Service Requests:** Service requests are established for parties not party sites. Within Teleservice, party sites are just used as ship-to, bill-to, or installed-at locations; the service request itself is associated with the party.
- **CRM Application Foundation User Management:** User registration and information maintenance functionality is available for parties not party sites. For example, in self-service, online registration, users can only register that they are employed by a certain company if the company is modeled as a party. Users cannot register as employed by an organization modeled as a party site.
- **Customer Interaction History:** Customer facing applications track customer interactions or touch points at the party and account layer not at the party site layer. For example, if you want to record that a customer from a particular organization called in, the interaction will not be recorded properly if the organization is modeled as a party site.

- **Notes:** Notes are created for parties not party sites. For example, a sales or service representative might want to record notes for their customers but would not be able to do so for customers modeled as party sites.
- **Tasks:** Tasks are recorded for parties not party sites. You cannot log a task, such as “call back this customer”, for a customer modeled as a party site, because all tasks are associated at the party level.
- **Prospect and Customer Classifications:** Oracle Trading Community includes a classifications model that allows you to categorize different parties into a pre-defined classifications scheme. You can adopt an industry standard such as NAIC, NACE or SIC or create a user defined classification that makes the most sense for your business. Currently only parties are classified so you cannot take advantage of this classification feature for business entities modeled as party sites.

The bottom line is that you should model a business entity as its own party if you want to “recognize” that particular business entity and take advantage of key E-Business Suite functionality. Party sites should only be created to identify valid locations for a party. If you want a business entity to be an active player in the trading community, the business entity should be its own party.

Note: realize that, if you create multiple parties for the different business entities of one company, then, in current release, you may not have direct visibility to certain activities for the company as a whole.

However, realize that if you create multiple parties for the different business entities of one company, in current releases, you may not have direct visibility to certain activities for the company as a whole. Although many applications have begun to introduce rollup functionality, some ERP applications have not yet introduced functionality to provide consolidated activity data for all parties linked by party relationships. For example, if you model Business World’s three business entities as their own parties linked together by party relationships, each would have its own leads and opportunities. Currently there is no rollup reporting that totals the leads and opportunities for Business World as a whole. Similarly, at this time you cannot roll up invoicing for the three Business World parties linked together by party relationships. Consolidated billing is done at the account site level and does not span across multiple parties linked by party relationships. And separate statements are created for each party and its associated account, so you would not be able to create statements for Business World as a whole. However, Oracle Receivables and Oracle Collections have introduced functionality in 11.5.9 that address many rollup gaps. For example, the Oracle Receivables team created a new Credit Management workbench that takes advantage of party relationship hierarchies in providing rollup functionality. They have enhanced the calculation of credit related data points for a party and its associated accounts such that you now have the option of considering the transactions of all child parties (in a given hierarchy) and the accounts for those parties. In addition, the Oracle Collections team has introduced functionality that allows implementing organizations to utilize “View By” functionality, which can show a rolled-up view of a customer across all accounts, including transactions associated with each account. Collections dunning plans, strategies and scoring also

leverage this roll-up view for a consolidated approach to managing customer delinquencies.

Although gaps do exist in the area of corporate roll-ups, it is important to note that many applications have already begun to introduce this functionality, and many more are expected to in future releases. An infrastructure, known as the “Get Children API”, which leverages the new hierarchy structure enforced in Relationship Manager, has been created that makes it possible to traverse a hierarchy in the E-Business Suite. In future releases, additional E-Business Suite applications will start using this API to produce roll-ups for transactions specific to their applications. For example, the Sales applications can use it to produce opportunities that span parties within a single corporate hierarchy. Until this happens, deploying companies can use this API, which is public, to create their own custom reports and views.

While accounts are addressed in a separate question (#2), the implications of creating accounts are interdependent with this party vs. party site decision. If you choose to create multiple parties for one company, and then proceed to create accounts for each of those parties in the second question, there will be a set of ramifications associated with creating those accounts (e.g. sharing of bill-to and ship-to information). Those implications may affect your original decision of how many parties to create. Therefore, we recommend that you answer the questions in order and then loop back to review each a second time to address the interdependencies.

Question 2

Now that you have identified the parties you need, for which of these parties do you need to create accounts?

How to Answer Question 2

Recommendation 1: Create an account for any party you have a selling relationship with. *Note: you may actually want to create more than one account for a single party. That decision is addressed in the following question (#3).*

- While a logical time for an account to be created is when an order is placed, you can have an account prior to that. For example, if you have already negotiated the shipping and billing terms that will be used if an order is ever placed, you can record these attributes of your relationship with the party as an account.
- Do not create accounts to represent “real things”. An account just represents the attributes of your selling relationship with some other real thing (e.g. Party). Axiom 5 is a good test: “An account should not need an account name to allow the deploying organization to identify the organization or person.” However, although you do not need a name for an account, the deploying company may wish to add an

Account Description so that they can distinguish between accounts, especially if a party has multiple accounts.

Implications of different answers to Question 2

There are two categories of implications for this question:

1. The first category includes the straight-forward implications of whether or not you create an account for a particular party. If you create an account, what do you get? If you don't create an account, what do you lose?
2. The second category includes the implications of decisions for question 1 and question 2 as they interrelate. If you create multiple parties for one company in question 1 and then create accounts for those parties in question 2, what is the overall result?

Straight-forward implications of having an account

Having an account allows you to record information about your selling relationship with a party, such as billing and shipping preferences, and enables you to accomplish transactional activities associated with your selling relationship. The following is a sampling of functionality for which an account is necessary:

- to place orders in Order Management
- to create invoices in Oracle Receivables
- to bill from contracts
- to bill for repairs
- to create service contracts
- to establish service installations

Implications of question 1 and question 2 considered in parallel

What are the implications of creating individual parties (with associated accounts) for different business entities within a single company rather than a single party and account to represent the company as a whole?

In general, identifying your customers in more granular units rather than as one big whole allows you to know them better and thus serve them better. One advantage of creating unique parties and associated accounts for different business entities is that you can tailor your relationship with each business entity to match its unique characteristics and requirements. By recording unique demographic information, billing terms, shipping terms, credit limits, etc. for each business entity, you can fine-tune your selling relationship and provide a superior customer experience. For example, by having different parties to represent an organization, you can target different marketing campaigns to each division (represented by a party), and additionally, have the ability to assign specialized sales representatives for each in order to maximize the sales effort. In addition, if one particular business entity has

a terrible payment history with you, you could set a low credit limit just for that particular business entity without simultaneously punishing the other business entities with that same credit limit.

Having separate parties and associated accounts for individual business entities also gives you visibility to activity such as orders, revenue, and accounts receivable per business entity. This added layer of visibility can help you profit from your customers more. As an example, let's say you model Business World as one party with one account and their transactional activity shows that they are an "ok" customer. They bring in a moderate amount of revenue and generally pay on time. You would probably devote a moderate amount of time and energy to this customer. Alternatively, let's say you model it as 3 parties linked by party relationships: an HQ and two subsidiaries. With separate accounts you would see activity separately and would be able to identify that subsidiary 1 is actually a great customer with extremely high revenue and a great payment record while subsidiary 2 is a terrible customer that returns most products and never pays on time. Furthermore, with separate parties to denote these three entities, you will have visibility into how many leads and opportunities are being qualified and turned into sales for each entity. With this additional insight you would know to expend much more of your energy on subsidiary 1 to maximize profitability from this customer as a whole. In addition, you will have the data points to analyze why deals are not being closed for the other entities, and can therefore focus on altering their customer experience in an effort to make them better customers in the future.

One thing to consider, though, is that in the current release not all applications have exposed rolled up activity for parties (and their associated accounts) in a hierarchy. The ability to expose and traverse hierarchies has been introduced by TCA and applications across the E-Business Suite are in the process of incorporating this functionality. You may be tempted to model a company as one party with multiple party sites (and thus one account and multiple account sites) purely because this approach can provide visibility to consolidated activity in a few functional areas. However, we recommend that you do not use this approach because these particular pieces of functionality are available for historical reasons only and they do not reflect the intentions of the new model or the future direction of Oracle Applications. Since all E-Business Suite applications are using the new, common Trading Community Architecture, they will have the opportunity to exploit the capabilities of the new model, such as party relationship hierarchies, as they add new features and improve existing ones for future releases.

Question 3

Should you create multiple accounts for any of the parties? Should you create multiple bill-to account sites for any given account?

How to answer Question 3

Recommendation 1: You need multiple accounts for one party if you market, sell to, and service that company as one entity, but have different selling relationships with that party.

To identify the relationships between external parties and your internal business entities, consider how you view your internal organization and how your company operates. For example, your company might be broken into very distinct product divisions that operate independently and are responsible for their own profit and loss. In that case, each of those divisions might have its own relationship with the same external party and thus would need its own account with that party. Or, you might be running a global organization that has subsidiaries in different countries that operate independently under separate management. In that case, you would want to create an account for each subsidiary that has its own relationship with the external party. Alternatively, perhaps you are a global company with locations in different countries but all those locations operate together as single global unit because management and operations are consolidated. In that case, you might only need one account for an external party.

Even within one internal business entity, however, you might negotiate different relationships with one external party. For example, if you are providing both phone and cable service to the same customer, you may have different terms and preferences for the cable service than for the phone service, so you could create multiple accounts to represent these different terms (i.e. relationships) with the same party.

In addition to your internal business structure driving the need to create multiple accounts per party, often times the end customer's structure drives the requirement. For example, you may have a customer with whom you interact with as one entity, but this organization purchases your products for their own personal use as well as for government project use. For their government purchases, GSA pricing exists, whereas when they purchase directly for their use, they have a globally established discounted price. In this case, you may want to create two accounts (one for GSA and one for direct) under the same party.

Recommendation 2: To determine whether you need more than one account, consider Axiom 6: "An account should typically have only one active bill-to site, given that most times, different terms and conditions apply to different bill-to sites". If you need an additional bill-to site, you probably need another account because you probably have a second relationship with that party. This might seem counter-intuitive because the data model does support the creation of multiple bill-to sites for a single account. As described above, if your organization encounters a situation where multiple bill-to addresses fall under the exact same terms and conditions of another bill-to address for a customer, you may indeed create multiple bill-to sites under one account (remember the lockbox scenario). Please note the following as an additional example of where multiple bill-to account sites can be used within the context of the TCA Best Practices model:

- Suppose an implementing organization sells widgets to Business Co., which is a global organization with locations across the world. Business Co. has established countrywide purchasing agreements with the implementing organization, whereby every location in each respective country receives the exact same discount and contractual agreements when making a purchase. As such, every site within each country is subject to the exact same financial terms and conditions. In addition, the implementing organization assigns one sales representative to all sites within each country, and all marketing efforts are targeted at the individual country's domestic HQ (e.g. Business Co. US or Business Co. UK). Finally, the implementing organization desires that all Service Requests and other CRM transactions be tied to the domestic HQ level for each of Business Co.'s country locations. Given the business model noted herein, it would be in line with TCA Best Practices to create one Party for each domestic HQ of Business Co. (e.g. Business Co. US), with multiple party sites, and associated account sites, to track the various locations within a particular country. In addition, each account site could be marked as both a bill-to and ship-to site given the fact that each of the bill-to sites noted herein can be used interchangeably. Essentially, since all bill-to sites for the Business Co. US party / account are subject to the exact same terms and conditions, and the implementing organization does not want to deal with any of those party / account sites in any specific fashion (e.g. track Leads, Opportunities, Service Requests, separately), then multiple bill-to account sites could be used to satisfy the business requirements.

As noted above, please keep in mind the business flow implications of creating multiple bill-to sites under one account. Generally speaking, for the purposes of keeping in line with the TCA Best Practices model and not veering back to the “way it was before”, we typically suggest modeling one active bill-to site per account. However, for cases where multiple bill-to account sites best suit the interactions with a particular customer, TCA Best Practices supports this model.

Implications of creating more than one account

Creating more than one account allows you to distinctly model different selling relationships with the same party and it enables you to see the activity of each relationship separately. On the flip-side, however, you currently will not have direct visibility to consolidated activity across all the accounts for one party. For instance, because Order Management has not yet fully exploited the potential of the Trading Community Model, existing Order Management functionality generally centers on how much business a single account provides and rollups are currently limited to the account level. As one example, the Comprehensive Order Report presents the number of orders that have been placed for a particular account; it does not provide a comprehensive total of orders for all accounts for a single party.

In addition, if multiple accounts exist, it may be difficult for Order Management users to identify the account with which they want to work given that the Account Description (formerly Account Name) field is not currently displayed on the Sales Order form in Order Management. However, as of 11.5.9, Quoting has introduced the Account Description field to provide users with the ability to properly distinguish amongst multiple accounts that exist for one party. Until this piece of functionality is introduced in Order Management, you may wish to establish a business process convention for users to follow to select the appropriate account if multiple accounts exist for one party.

Question 4

Should you create any account relationships?

How to Answer Question 4

Recommendation 1: Account relationships should be created when you want to establish a relationship between two accounts such that one account can pay for another account's invoices or one account can be used as the ship-to for another account's orders.

- Account relationships should not be used to construct the organizational hierarchy of different business entities for one company.

In determining the best prospect and customer model for your organization, it is important not to take any particular recommendation to an extreme. The more you learn and understand about TCA, the more comfortable you will feel with making decisions that will allow you to model your data in the best way possible way for your organization. Again, the recommendations outlined herein are general recommendations, that are sound in concept and principle, and as noted above, each scenario will depend on the deploying company's interaction points with their customers.

APPLYING QUESTIONS AND AXIOMS TO SAMPLE DATA

Below we will walk through the key questions for four sets of example data. Each set of example data does not have information relevant to all four key questions, but each of the questions will be addressed within at least one set of data.

Example Prospect and Customer Data

For all four sets of data, assume the following scenario:

You, the company deploying Oracle E-Business Suite, sell office supplies to dealers and direct customers. You also supply temporary personnel; companies can contract with you for the placement of temporary office managers. You have the organizational structure shown below. While you have business entities in France, Italy, and Brazil, you currently operate as one global company because your management and operations are consolidated.

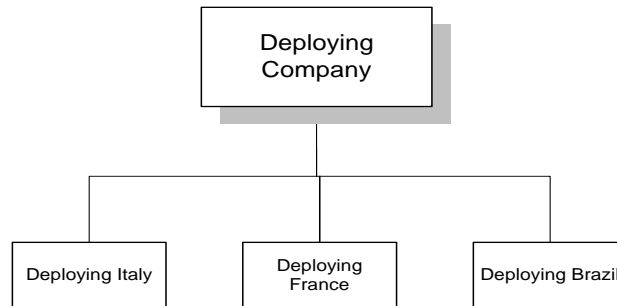


Figure 5: Organization structure for example deploying company

Assume you have the following customer and prospect data:

Example #1: Hilman and Associates

You just purchased information about Hilman and Associates from a marketing agency. Hilman and Associates is an entertainment company that you have never heard of before. The organizational structure is shown below. The different business entities (Hilman N.Y., Hilman Los Angeles, and Hilman Oakland) manage different media channels (television, motion pictures, and print) so you will use different marketing strategies and campaigns to target the different business entities. Your top corporate initiative at this time is to improve customer service. To help evaluate your current level of customer service you want to track your customer interactions and touch points with Hilman N.Y., Hilman Los Angeles, and Hilman Oakland separately. According to the marketing agency supplying the data, Hilman NJ is not associated with a media channel. You assume this means they do not generate business from this site and thus you are not planning to market to this business entity.

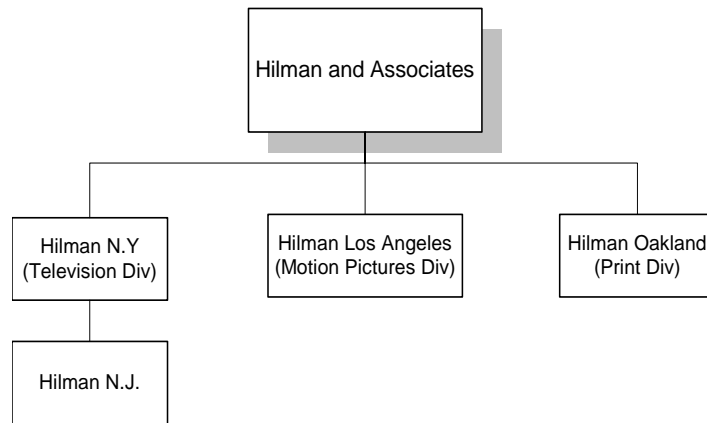


Figure 6: Organization structure for Hilman and Associates Example

Question 1

When considering Hilman and Associates (including the multiple business entities that make up its organizational hierarchy), which entities should be modeled as parties? Which should be modeled as party sites?

Sample answer

You identified that you want to interact with the different business entities individually because you intend to use different marketing strategies and campaigns to target Hilman N.Y., Hilman Los Angeles, and Hilman Oakland. Since you are interacting with them distinctly, each should be its own party. In fact, since marketing functionality drives off parties not party sites, you will not be able to market to a business entity separately if it's not set up as its own party. In addition, customer interactions are recorded for parties so you can only track interactions separately for the three business entities if they are set up as their own parties. Finally, it would not make sense to model these as party sites of one Hilman and Associates party because party sites are simply meant to represent the valid set of locations for a party (axiom #2). Since you are not planning to interact with Hilman NJ, you do not need to model that business entity. If you later want to interact with the Hilman NJ business entity, then you could model it as a party at that time. After creating separate parties for Hilman N.Y., Hilman Los Angeles, and Hilman Oakland, you could link those parties into the Hilman and Associates organization structure using party relationships to create a corporate hierarchy.

While creating parties for the different Hilman business entities will provide functionality and visibility at the business entity level, it will not currently give you visibility to your rolled up activity with all Hilman parties combined. For example,

you would not be able to see consolidated totals for activity such as opportunities, quotes, or service requests for all the Hilman business entities linked together by party relationships.

Question 2

Now that you have identified the parties, for which parties do you need to create customer accounts?

Sample answer

Because you have never heard of Hilman and Associates, you clearly do not have a selling relationship with them yet. Therefore, you do not yet need accounts for the parties you created for the Hilman N.Y., Hilman Los Angeles, and Hilman Oakland. However, if you are going to market to them with campaigns, they very well may become a customer for you at a later point. As soon as you need to record attributes about a selling relationship with one of the parties, an account would be needed for that party to store the relationship information.

Question 3

Should you create multiple accounts for any of the parties? Should you create multiple bill-to account sites?

Sample answer

There is no indication in the example data that you need to create any accounts, let alone multiple accounts, for any of the parties. Multiple accounts per party may come into play sometime in the future whereby the Motion Pictures division of Hilman and Associates established two separate purchasing agreements with your company, but you still market, sell to, and service the organization as one entity.

Question 4

Should you create any account relationships?

Sample answer

Because no accounts have been created, there is no need to consider account relationships.

Note that if the parent company, Hilman and Associates, establishes a corporate purchasing agreement with your organization, this pricing agreement should be established at the parent company level (Hilman and Associates) and linked to the subsidiary business entities via Account relationships.

Example #2: Bigmart

Bigmart is a retail organization with an HQ in Texas and three stores nationwide as shown in the organizational structure below. You assign different sales reps to sell office supplies directly to the individual stores based on geographic territories (east,

west, central). You want visibility to purchasing activity (orders, revenue, accounts receivable, aging, etc.) per individual store because you want to use the profitability of different geographic regions to help you decide where to open new stores of your own. Occasionally Bigmart's bills are paid by their parent company, Bigmart Conglomerate, so for some orders you must invoice Bigmart Conglomerate directly.

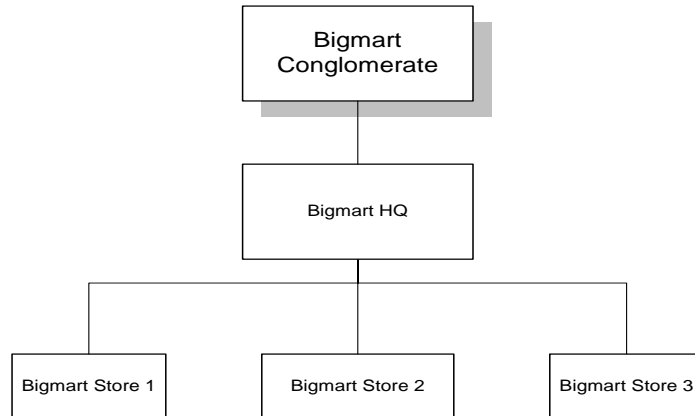


Figure 7: Organization structure for Bigmart example

Question 1

When considering Bigmart (including the multiple business entities that make up its organizational hierarchy), which entities should be modeled as parties? Which should be modeled as party sites?

Sample answer

According to your business practices, you want to distinguish between your activities with the different stores. In other words, you want to interact with Bigmart at the store level, so you would want to model each store as its own party. In addition, you want to assign different sales representatives to manage the opportunities for the different stores. You cannot do this unless each store is modeled as its own party because sales rep assignments are done at the party level not the party site level. While you want to distinguish between the different stores, you still want to know that they are part of the same company so you could use party relationships to construct the organizational hierarchy.

Question 2

Now that you have identified the parties, for which parties do you need to create customer accounts?

Sample answer

According to the data, you are already selling to Bigmart so by definition you need at least one account to represent your selling relationship with Bigmart. Because you created three different parties to represent the three stores, you can create an

account for each party so that the account-related activity, such as accounts receivable and aging, are tracked separately for the three stores. This will allow you to do the store profitability comparisons that your business practices require. In addition, because you want to bill Bigmart Conglomerate for some of the orders, you would also create an account for Bigmart Conglomerate (see question #4 below).

Also, as indicated earlier in this document, if you set up the stores as individual parties, in the current release of the E-Business Suite you will now have some direct visibility to consolidated activity for Bigmart as a whole. For example, you will have the aging for each store as well as a consolidated aging for Bigmart company as a whole.

Question 3

Should you create multiple accounts for any of the parties?

Sample answer

The information in this example does not suggest that you would need to create multiple accounts for Bigmart. Even though you have business entities in Germany, France, and Italy, you operate as one global unit so you would not need to create separate relationships (accounts) between your different business entities and Bigmart. And, there is no other information about Bigmart or the products being sold that suggests the need for setting up multiple relationships.

Question 4

Should you create any account relationships?

Sample answer

Since some of Bigmart's bills are paid by their parent company, Bigmart Conglomerate, you need to create account relationships between Bigmart Conglomerate and the other business entities so you can invoice Bigmart Conglomerate for purchases made by the rest of Bigmart.

Example #3: Business World

You have been selling office supplies to the HQ of Business World for three years. However, for the first time, they just placed an order for temporary personnel. They are contracting two office managers for one year. So far you have not interacted with any of the subsidiaries; all agreements and purchases have been made with HQ. You purchased Dun & Bradstreet data for Business World and received the D-U-N-S Numbers as shown on the corporate hierarchy. Because you often rely on information supplied by Dun & Bradstreet, you would like to set up your data to complement D&B's approach when possible and use the D-U-N-S Number as an identifier in your business practices.

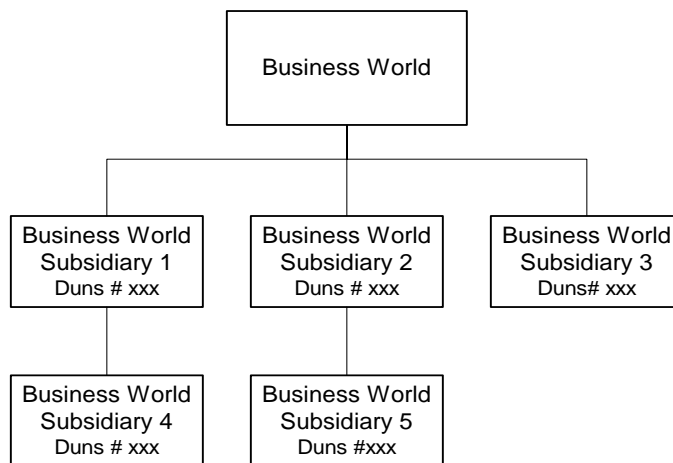


Figure 8: Organization structure for Business World example

Question 1

When considering Business World (including the multiple business entities that make up its organizational hierarchy), which entities should be modeled as parties? Which should be modeled as party sites?

Sample answer

Because your business practice is to use Dun & Bradstreet information and the D&B D-U-N-S Number as a unique identifier, *if you need* to model a particular business entity that has its own D-U-N-S Number, you should model that business entity as its own party rather than as a party site.

However, the first consideration should not be *how* you should model the business entity, but whether you actually need to model the business entity at all. You clearly interact with the HQ because all of your agreements and purchases are made with that unit. Because you interact distinctly with that business entity, and because you want to identify it by its D-U-N-S Number, the HQ business entity should be modeled as its own party. In the course of working with the HQ business entity, you may hear of several addresses associated with that business entity. If you want to distinguish the particular address used by D&B to identify the business unit, that address should be modeled in TCA as the identifying address for the party.

Because you don't interact with the subsidiaries you do not necessarily need to create parties for the subsidiaries. However, if you decide that you want to interact at the subsidiary level in the future, you could create parties for those units. By creating them as separate parties you can market to them individually and you can use the D-U-N-S number to identify them. If you decide to create parties for the subsidiaries, you can link those parties to the HQ party using party relationships.

Question 2

Now that you have identified the parties, for which parties do you need to create customer accounts?

Sample answer

Because you are selling to the HQ, you need an account with the HQ to record your selling relationship with that party. If you did create parties for the other subsidiaries, you would only need to create accounts for those parties if and when you establish a selling relationship with them (i.e., if they become your “customer” or if HQ makes a purchase that needs to be shipped to a subsidiary).

Question 3

Should you create multiple accounts for any of the parties?

Sample answer

Because you operate as one global unit, you probably do not need to create separate accounts between Business World and your different internal business entities such as Deploying Italy or Deploying France. However, you are selling two very different “products” to Business World HQ, office supplies and temporary personnel services. It is very likely that you have negotiated different agreements, or relationships, with Business World to handle the two different products. For example, the terms for the office supplies might be net-30 but for the contracted office managers it might be bi-weekly paychecks. If your relationship for selling office supplies is very different from your relationship for selling personnel services, you will want to create two separate accounts for the Business World party to record those relationships distinctly.

The implications of having multiple accounts have been addressed previously in the document. Basically, you gain the ability to track your business activity for office supplies separately from your business activity for temporary services. Conversely, at this time you will have some, but not complete visibility to total account activity for a particular Business World party.

Question 4

Should you create any account relationships?

Sample answer

There is no indication in the example data that you would need to create account relationships.

Example #4: Computer Service and Rentals

As shown in the organizational structure below, Computer Service and Rentals has a Computer Services division and a Computer Rentals division. Both divisions are located at the same address in San Jose, California. You sell supplies to both the

Computer Services division and the Computer Rentals division, but you establish separate contracts with the two divisions. You see potential for expanding your product line with both divisions so you are allocating additional sales resources to both. However, because you think selling to a service organization should be handled differently from selling to a rental organization, you assigned different sales teams to manage the leads and opportunities at the different divisions.

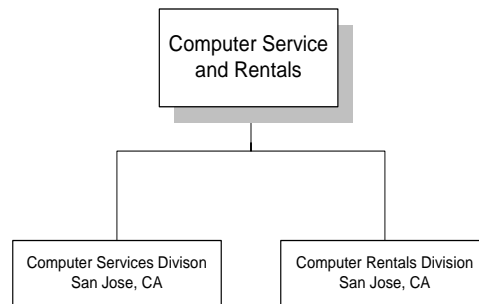


Figure 9: Organization structure for Computer Service and Rentals example

In addition, you, the deploying company, recently had a management shift that altered your company's operational strategy and distribution of power. Whereas you previously operated as one global, consolidated unit, the three country divisions now operate independently and are responsible for their own profitability. You now view your internal organization as three relatively autonomous business entities: Deploying France, Deploying Germany, and Deploying Brazil.

Question 1

When considering Computer Service and Rentals (including the multiple business entities that make up its organizational hierarchy), which entities should be modeled as parties? Which should be modeled as party sites?

Sample answer

Even though the Computer Services Division and the Computer Rentals Division are both at the same location, the business practice information suggests you should model the divisions as their own parties. To have separate contracts for the two business divisions, they must be created as their own parties rather than as party sites of one Computer Service and Rentals party because contracts are made for party and account combinations, and not for party sites. In addition, they need to be separate parties because you want to assign one sales team to the leads and opportunities for the Computer Services Division and another team to the leads and opportunities for the Computer Rentals Division and you can't do this unless they are both modeled as parties. And, by creating them as parties you will be able to use distinct marketing strategies to target each uniquely.

Question 2

Now that you have identified the parties, for which parties do you need to create customer accounts?

Sample answer

You already have a selling relationship with the two parties so you need an account to represent each of those relationships. The implications of having these accounts are the same as those explained in previous examples.

Question 3

Should you create multiple accounts for any of the parties?

Sample answer

Because your company now operates as three relatively autonomous business entities (Deploying France, Deploying Germany, and Deploying Italy), each of these business entities would have its own selling relationship with Computer Service and Rentals. If your Italian, French, and German business entities want to interact with both the Computer Services Division and the Computer Rentals Division, you would create six accounts total. You would need three accounts for the party representing the Computer Services Division (one for Deploying France, one for Deploying Germany, and one for Deploying Italy) and three accounts for the party representing the Computer Rentals Division. Of course, it might be that not all three of your business entities want to interact with both divisions of Computer Service and Rentals. For example, Deploying France may want to interact only with the Computer Services Division, in which case an account would not be needed for Deploying France and the party representing the Computer Rentals Division. In any case, the implications of having multiple accounts for the same party would be the same as described for the previous examples.

Question 4

Should you create any account relationships?

Sample answer

There is no indication in the example data that you would need to create account relationships. Although, if the Computer Service and Rentals parent company establishes a global purchasing discount with your organizations, or if they decide to purchase centrally on behalf of the other divisions, account relationships would be required.

CONCLUSION

While many considerations will influence your set up decisions, a select few are the most consequential. The considerations all center on an analysis and understanding of the following:

- how your internal company is organized and how it operates
- how the external customer or prospect is organized and how they operate
- how your company interacts, or intends to interact, with the external customer or prospect

A thorough analysis in these areas and an accurate understanding of the Trading Community Architecture entities, axioms, and intentions, will provide obvious conclusions to many set up choices and steer your decisions towards the optimal set up for your unique business.

Finally, for any data set up to be effective in the long run, clear and consistent business practices should be implemented to complement your data modeling decisions.



Oracle Trading Community Architecture Best Practices V.2
October 2003
Authors: Renee Connors and Adam Stafford

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com

Copyright © 2003, Oracle. All rights reserved.
This document is provided for information purposes only
and the contents hereof are subject to change without notice.
This document is not warranted to be error-free, nor subject to
any other warranties or conditions, whether expressed orally
or implied in law, including implied warranties and conditions of
merchantability or fitness for a particular purpose. We specifically
disclaim any liability with respect to this document and no
contractual obligations are formed either directly or indirectly
by this document. This document may not be reproduced or
transmitted in any form or by any means, electronic or mechanical,
for any purpose, without our prior written permission.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective owners.